US030430

PA 1122137

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

February 03, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/515,319* ✓
FILING DATE: *October 29, 2003* ✓

IB/04/52235

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

P. SWAIN
Certifying Officer

Please type a plus sign (+) inside this box → ☒

PTO/SB/16 (02-01)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a requ st f r filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

Express Mail Label No. EV 312 069 644 DATE OF DEPOSIT: 29 OCTOBER 2003

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
MARTEN	VAN DIJK	Cambridge, MA, US

☐ *Additional inventors are being named on the* _____ *separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

PROFILE MATCHING WITH APPLICATIONS IN BIOMETRICS AND PHYSICAL RANDOM FUNCTIONS

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☒ Customer Number 24737 → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☐ Firm or Individual Name	PHILIPS ELECTRONICS NORTH AMERICA CORPORATION
Address	580 WHITE PLAINS ROAD
Address	

City	TARRYTOWN	State	NY	ZIP	10591
Country	USA	Telephone	914-333-9627	Fax	914-332-0615

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages*	34	☐ CD(s), Number	
☐ Drawing(s) *Number of Sheets*		☐ Other (specify)	
☐ Application Data Sheet. See 37 CFR 1.76			

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ Applicant claims small entity status. See 37 CFR 1.27.

☐ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 14-1270

☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($) 160

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____.

Respectfully submitted, SIGNATURE		Date	29 OCTOBER 2003
TYPED or PRINTED NAME	DICRAN HALAJIAN	REGISTRATION NO. *(if appropriate)*	39,703
TELEPHONE	914 333-9607	Docket Number:	US030430

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C., 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Profile Matching with Applications in Biometrics and Physical Random Functions

Marten van Dijk*

Philips Research Laboratories

Abstract. We consider the problem of secret key agreement between two legitimate users whose profiles match sufficiently. We give practical solutions based on Reed-Solomon (RS) codes and one-way functions. We prove their security and analyze the leakage of information to an adversary who has at most a given finite amount of computational power. We show how biometrics and physical random functions can make use of these solutions. In particular, both an optical physical random function with a post-hash and a silicon physical random function can be made reliable and provably secure.

1 Introduction

In this paper we address the following situation (see also Maurer [M91,M93]):

- There is a physical system P which gives profiles to different principals. In particular Alice receives a profile A, Bob receives a profile B, and Eve receives a profile E.
- We want a protocol in which Alice is able to share a secret key K with Bob if Bob's profile B is close enough to Alice's profile A, that is, if the distance $d(A, B)$ is less than a certain threshold. The protocol may use public (broadcast) communication.
- The protocol should provide *authenticity* in the sense that
  - Alice knows that any person with whom she successfully shared a secret key has knowledge of a profile which is close to her profile, and
  - Bob knows that any person who is successfully sharing a secret key with him has knowledge of a profile which is close to his profile.
- The protocol should be *secure* in the sense that anyone (Eve) who obtained a profile (E) from P with large distance to A should only obtain a negligible amount of information about K.

We notice that if $d(A, B)$ is defined as the mutual information between A and B then this situation is like the example in [M91,M93] where the physical system P plays the role of a satellite broadcasting a random binary string X and where the profiles are noisy versions of X received by Alice, Bob, Eve, and others.

* Visiting scientist at the MIT Computer Science and Artificial Intelligence laboratory (CSAIL), 200 Technology Square, Cambridge, USA. Email: marten@mit.edu.

If we allow the security to be based on a computationally difficult problem, then a straightforward solution is to first use public communication to set up a secure channel between Alice and Bob and to share a secret key K (security), for example by using Shamir's no-key algorithm [MvOV96], and secondly to use the secure channel for a profile matching protocol (authenticity), that is a protocol which convinces both Alice and Bob that their profiles sufficiently match. We will allow computational security, but our goal is to design an efficient solution where security and authenticity are simultaneously achieved.

We give a first solution based on Reed-Solomon (RS) codes in Section 3.1 and prove its security. Even though the fuzzy commitment scheme [JW99] has lots in common, we show that it is in general less secure. In Section 3.2 we introduce a technique to use a one-way function to create an erasures channel. This leads to a perfect scheme[1] in the sense that one is either able to reconstruct the key K or does not obtain any information about K. The solutions presented so far only require a single public communication from Alice to Bob. In Section 3.4, we explain how in combination with a one-way function public feedback from Bob to Alice can be used. This leads to a simple scheme in which a minimal amount of information is leaked by Alice and Bob to one another and to others. The solution also offers the possibility to match unordered sets, which leads to a more efficient scheme than the one published in [JS02].

In Section 4 we describe the relationship with unconditional secure key generation (see for example [CK78,M93,AC93]). In particular we prove a relationship between the entropy of Eve about the secret key K and the amount of computing power of Eve if one-way functions are used. In Section 5, we analyze explicit examples and we show how the parameters of the RS code in our solutions can be computed.

In Section 6.1 we explain an application towards biometrics [LT03]. In Section 6.2 we discuss physical random functions and we show how our solutions for profile matching make physical random functions reliable and secure.

2 Model

We are interested in the model of Figure 1. A physical system P gives a profile A to Alice and a profile B to Bob. We address the problem of how Alice can use her profile to transmit a secret key K to Bob such that Bob can only recover K if his profile is close to Alice's profile, that is $d(A,B)$ is small. Anyone (Eve) who obtained a profile (E) from P with large distance to A can only obtain a negligible amount of information about K. Summarizing,

$$\forall_B\, [d(A,B) \text{ small } \Rightarrow K \text{ can be recovered from } (I,B)]$$

and

$$\forall_E\, [d(A,E) \text{ large } \Rightarrow \text{ it is infeasible to obtain from } (I,E) \text{ information about } K]$$

[1] Perfect as in perfect secret sharing schemes.

Notice that these requirements are not formulated by using concepts from information theory. This is because we allow the security to be based on a computationally difficult problem. In particular, we will use a one-way (hash) function[2] $h(.)$. If we remain in the information theoretical setting and if we define the distance $d(A,B)$ to be the mutual information $I(A;B)$ between A and B then we are in the situation of [M91,M93].

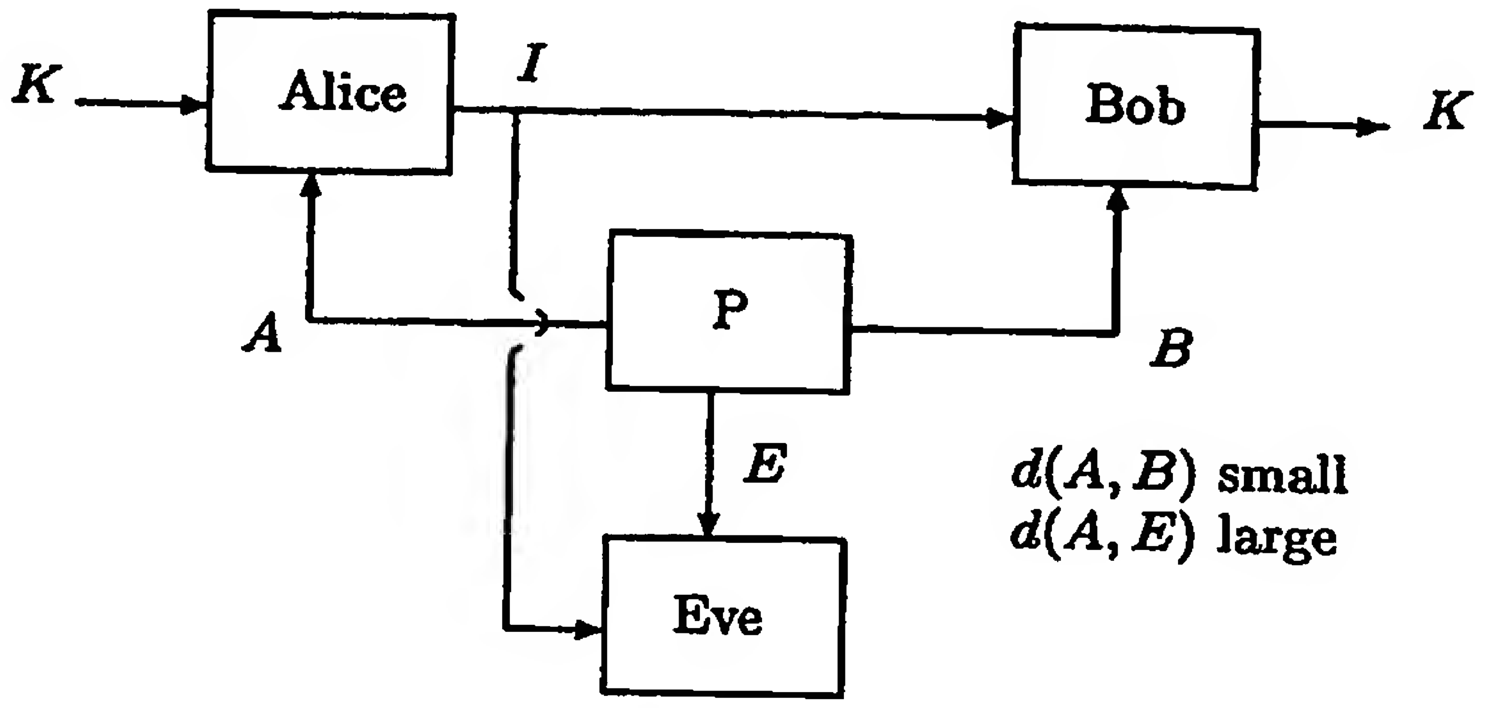

Fig. 1. Profile matching model.

As soon as Bob has recovered a secret key K', he needs to convince himself that $K = K'$. Equality ($K = K'$) should imply that his profile is close to Alice's profile (inequality should imply that Bob's profile, just like Eve's profile, has a large distance to Alice's profile and that K' does not reveal information about K). As in operating systems, which store hashes of passwords instead of storing passwords explicitly [FK89,MT79], a straightforward solution is to include $h(K)$ in I, which is a commitment of K. If Bob's profile is close to Alice's profile then $K = K'$ and $h(K) = h(K')$. If his profile has a large distance to A then, according to our requirements, $K \neq K'$ and $h(K) \neq h(K')$. Of course a simple CRC check instead of a one-way function $h(.)$ suffices for this purpose. However, a CRC check of K reveals information about K to any Eve who receives I over the public channel from Alice to Bob.

As soon as Alice and Bob share a secret key, they can use this key in cryptography primitives to exchange sensitive data. Of course Alice and Bob only know that their profiles match enough. They do not necessarily know who the other person is. Often Alice or Bob only require that the other person is legitimate in the sense that their profiles match and that one of them is physically linked to the physical system P. For example in the biometrics application of Section

[2] A one-way function is a function $h(.)$ such that for each x in the domain of h, it is easy to compute $h(x)$, but for essentially all y in the range of h, it is computationally infeasible to find any x such that $y = h(x)$ [MvOV96].

6.1 the physical system P is a device which measures Alice's fingerprint. In the smartcard application of Section 6.2 the smartcard plays the role of Bob and contains the physical system P.

3 Efficient Protocols

Let

$$A = (a_1, \ldots, a_n),\ B = (b_1, \ldots, b_n),\ \text{and } E = (e_1, \ldots, e_n)$$

be the profiles of Alice, Bob, and Eve. In this section we present solutions for profile matching where we use the Hamming distance,

$$d_H(A, B) = |\{i : a_i \neq b_i\}| \text{ and } d_H(A, E) = |\{i : a_i \neq e_i\}|.$$

3.1 A Solution using RS codes

See Figure 2, Alice wants to share the key $K = (K_1, \ldots, K_k)$ with Bob. To this purpose Alice uses an

$$[n + k + d - 1, n + k, d]$$

Reed-Solomon (RS) code[3] over $GF(q)$ (this requires $q - 1 \geq n + k + d - 1$). The RS code is systematic which means that any $n+k$ positions form an information set[4]. The minimum distance is equal to d, in particular this implies that any set of $d - 1$ erasures[5] can be corrected. This also follows from the fact that the non-erasure positions form an information set. This means that Alice is able to encode the $n + k$ positions

$$(K_1, \ldots, K_k, a_1, \ldots, a_n)$$

into a RS code word

$$W = (K_1, \ldots, K_k, a_1, \ldots, a_n, p_1, \ldots, p_{d-1}).$$

The entries p_i are called parities and are transmitted by Alice to Bob.

Bob uses the $d - 1$ parities to construct the vector

$$(?, \ldots, ?, b_1, \ldots, b_n, p_1, \ldots, p_{d-1}),$$

the first k entries are question marks representing erasures. Compared to the code word W, the constructed vector has k erasures and $d_H(A, B)$ errors ($a_i \neq b_i$ for $d_H(A, B)$ unknown positions i). Since the RS code has minimum distance d, a

[3] The triple $[n+k+d-1, n+k, d]$ denotes length, dimension, and minimum distance of the RS code.

[4] For each selection of entries at the positions in an information set there exists a unique corresponding code word.

[5] Erasures are positions for which no entry is received or computed.

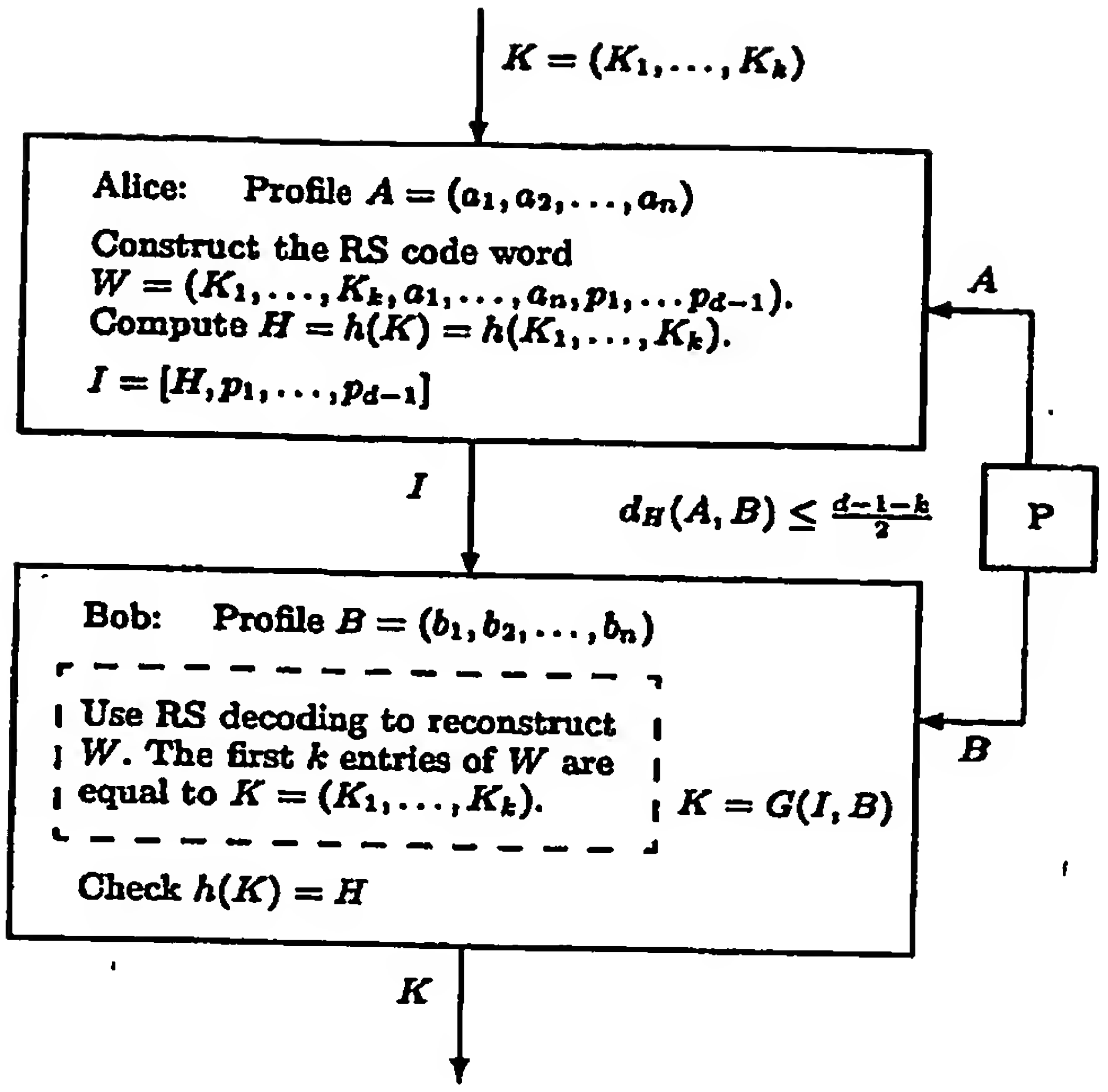
$K = (K_1, \dots, K_k)$
Alice: Profile $A = (a_1, a_2, \dots, a_n)$
Construct the RS code word
$W = (K_1, \dots, K_k, a_1, \dots, a_n, p_1, \dots p_{d-1})$.
Compute $H = h(K) = h(K_1, \dots, K_k)$.
$I = [H, p_1, \dots, p_{d-1}]$
A
I
$d_H(A, B) \leq \frac{d-1-k}{2}$
P
Bob: Profile $B = (b_1, b_2, \dots, b_n)$
Use RS decoding to reconstruct W. The first k entries of W are equal to $K = (K_1, \dots, K_k)$.
$K = G(I, B)$
B
Check $h(K) = H$
K

Fig. 2. RS-Protocol: Profile matching by using a Reed-Solomon code.

classical errors-and-erasures decoding algorithm [MS77] corrects the constructed code word to W if and only if

$$k + 2d_H(A, B) \leq d - 1.$$

This proves that Bob can reconstruct W, in particular K, if

$$d_H(A, B) \leq (d - 1 - k)/2. \tag{1}$$

The more advanced (and less efficient) Guruswami-Sudan algorithm [GS99] reconstructs the code word W if

$$d_H(A, B) < (n + d - 1) - \sqrt{(n + d - 1)(n + k - 1)}.$$

For example, see Section 5, for $n = 5794$, $d = 4157$, and $k = 442$, the Guruswami-Sudan algorithm decodes up to $d_H(A, B) \leq 2073$ while a classical errors-and-erasures decoding algorithm decodes up to $d_H(A, B) \leq (d - 1 - k)/2 = 1857$.

The $d - 1$ parities are transmitted over a public channel. This means that the knowledge of Eve can be represented by the vector

$$(?, \dots, ?, e_1, \dots, e_n, p_1, \dots, p_{d-1}).$$

To prove that this vector does not contain any information about K, we show that this vector corresponds to each possible key equally likely.

Let $K' = (K'_1, \dots, K'_k)$. Without loss of generality, let the first $l = n - d_H(A, E)$ entries of E be equal to the first l entries of A. Let us assume that *given* $d_H(A, E) = l$, *Eve's profile* E *is uniformly distributed over* $\{X : d_H(A, X) = l\}$. By our assumption on the distribution of E, the last $n - l$ entries of E do not contain any information about A; to Eve, the last $n - l$ entries of A are uniformly distributed. Suppose that

$$k + n - d_H(A, E) + d - 1 = k + l + d - 1 \leq n + k$$

or equivalently $d_H(A, E) \geq d - 1$. Then there exists a code word

$$W' = (K'_1, \dots, K'_k, a_1, \dots, a_l, a'_{l+1}, \dots, a'_n, p_1, \dots, p_{d-1})$$

for some $a'_{l+1}, \dots, a'_n$ because the positions outside $a'_{l+1}, \dots, a'_n$ are part of an information set. To Eve, the last $n - l$ entries of A are with uniform probability equal to $a'_{l+1}, \dots, a'_n$. This proves that to Eve K is with uniform probability equal to K'. That is, Eve does not obtain any information about K if

$$d_H(A, E) \geq d - 1 \tag{2}$$

The implication also holds in the other direction. (If $d_H(A, E) \leq (d - 1 - k)/2$ then Eve can reconstruct K and if $(d - 1 - k)/2 < d_H(A, E) < d - 1$ then Eve can reconstruct partial information about K.)

Inequalities (1) and (2) prove the following lemma.

Lemma 1. *Suppose that, given* $d_H(A, E) = l$, *Eve's profile* E *is uniformly distributed over* $\{X : d_H(A, X) = l\}$. *Then in the protocol in Figure 2 there exists a distance* d *such that Bob can reconstruct* $K = (K_1, \dots, K_k)$ *and Eve does not obtain any information about* K *if and only if* $d_H(A, B) \leq (d_H(A, E) - k)/2$.

Let us compare our construction with the fuzzy commitment scheme in [JW99] for RS codes, which works as follows. Alice uses an $[n, n-d+1+k, d-1-k]$ Reed-Solomon (RS) code over $GF(q)$ (this requires $q - 1 \geq n \geq n - d + 1 + k$). The RS code is systematic meaning that any $n - d + 1 + k$ positions form an information set. Alice chooses random entries $R_1, \dots, R_{n-d+1}$ and she encodes

$$(K_1, \dots, K_k, R_1, \dots, R_{n-d+1})$$

into a RS code word W. Alice transmits $W + A$ to Bob. Bob constructs $W + A - B$. Notice that W and $W + A - B$ agree on $n - d_H(A, B)$ positions. By using an errors-only decoding algorithm, Bob reconstructs W if and only if $2d_H(A, B) \leq d - 1 - k$, that is $d_H(A, B) \leq (d - 1 - k)/2$.

By using a similar argument as before, we want to prove that Eve does not obtain any information about K if and only if $k + n - d_H(A, E) \leq n - d + 1 + k$, that is $d_H(A, E) \geq d - 1$. We need to be careful. Suppose that A is uniformly distributed. Then Eve can construct $W + A - E$ and discard

$W+A$ and E without losing any information about W and in particular K (see Section 4 for more explanation). Eve does not know which and how many positions in A and E match. We can prove that if $d_H(A,E) \geq d-1$ then for any $K' = (K'_1, \ldots, K'_k)$ there exists a code word $W' = (K'_1, \ldots, K'_k, \ldots)$ such that $d_H(W+A-E, W') \geq d-1$. In this sense W and W' with the corresponding K and K' are equally likely. However, we may find a list of W' for which $d_H(W+A-E, W')$ is minimized. These W' may have a significant probability to be equal to W in which case information about K is leaked to Eve. The Guruswami-Sudan algorithm [GS99] produces such lists[6]. It is unclear what the security of the fuzzy commitment scheme is in the profile matching model. Besides the difference in security, another difference with the protocol in Figure 2 is that Alice transmits more symbols in $GF(q)$.

Both the fuzzy commitment scheme and our construction can be used with arbitrary codes. For example the code may be a multi-dimensional vector space over the real numbers. In other words, the code words represent points in a lattice. In this setting the construction of [JW99] leads to the quantized index modulation (QIM) method of [CW01] used in watermarking.

As a final remark we notice that the assumption on the distribution of E may not hold. If it does not hold Eve may try to obtain some information about K by using soft decision decoding of RS codes [KV00,PV03,KJV+03] which is based on list decoding as developed in [GS99]. The following lemma gives a bound on Eve's knowledge for arbitrary distributions that describe how the profiles A, B, E, etc. are generated by the physical system P.

Lemma 2. *In the protocol in Figure 2, Eve's knowledge about the secret K is bounded by*[7]

$$I(KA;EP) \leq I(A;E) + H(P),$$

where $I(A;E)$ is the mutual information between the profiles A and B and where $H(P) \leq d-1$ is the information contained in the parities.

Proof. We derive

$$I(KA;EP) = H(EP) - H(EP|KA) = H(EP) - (H(P|KA) + H(E|KAP)).$$

Notice that AK uniquely defines the parity information described by P, hence, $H(P|KA) = 0$. Furthermore $KP \to A \to E$ forms a Markov chain, which implies that $H(E|KAP) = H(E|A)$. By combining all equations, we obtain

$$\begin{aligned} I(KA;EP) &= H(EP) - H(E|A) = H(E) + H(P|E) - H(E|A) \\ &= I(A;E) + H(P|E) \leq I(A;E) + H(P). \end{aligned}$$

□

[6] In particular, Guruswami and Sudan proved that the minimizing W' is unique if $d_H(A,E) \leq n - \sqrt{n(n-d+k)}$ (which is $< d-1$).

[7] We refer to [CT91] for the definitions of (conditional) entropy $H(.|.)$ and (conditional) mutual information $I(.;.|.)$.

Let A be uniformly distributed, let $d_H(A,E) = l$ be fixed, and suppose that Eve's profile E is uniformly distributed over $\{X : d_H(A,X) = l\}$. Then $I(A,E) = n - l$ and by Lemma 2,

$$k + n - H(KA|EP) = H(KA) - H(KA|EP) = I(KA;EP) \leq n - l + d - 1,$$

which proves $H(KA|EP) \geq k + l - (d-1)$. This implies that if $l \geq d-1$ then $H(KA|EP) \geq k$ and Alice and Bob can use K and A to distill k symbols which are secret to Eve (who only knows E and P). This corresponds to Lemma 1.

For arbitrary distributions, Alice and Bob need to extend the protocol in Figure 2. In general, Alice and Bob cannot use K as a secret key. Alice and Bob should compress KA to $s = H(KA|EP) \geq H(KA) - (I(A;E) + H(P))$ secret key bits by using privacy amplification [C97]. For example, multiplication with a random binary $(k+n) \times s$ matrix leads to a key of s bits about which Eve has only negligible information [CH77]. Of course, Alice and Bob can also use a hash function [MvOV96].

3.2 A Perfect Protocol by using One-Way Functions

See Figure 3, we propose to use the one-way (hash) function $h(.)$ not only for a commitment towards K but also to create an erasures channel as we will explain. The protocol is the same as in Figure 2 except that Alice also transmits the vector

$$(h(a_1), \ldots, h(a_n)).$$

As we will see the one-way function $h(.)$ in this vector is used to create an erasures channel.

Bob computes $(h(b_1), \ldots, h(b_n))$ and compares $h(a_i)$ and $h(b_i)$. This leads to the set

$$S = \{i : h(a_i) = h(b_i)\} = \{i : a_i = b_i\},$$

where the second equality holds with overwhelming probability by the definition of one-way (hash) functions. Notice that $|S| = n - d_H(A,B)$. Together with the $d-1$ parities which Bob received, he knows the entries of $n - d_H(A,B) + d - 1$ positions in the code word W. The other positions are regarded as erasures. By using erasures-only decoding Bob retrieves the whole code word if and only if the known positions contain an information set (any $n+k$ positions), that is,

$$d_H(A,B) \leq d - 1 - k. \tag{3}$$

In particular, Bob reconstructs K.

Eve may also compare $h(a_i)$ and $h(e_i)$. If they are equal she knows that $a_i = e_i$. If they are not equal then she obtains at most a negligible amount of information about a_i by the definition of one-way functions. Let us again assume that *given* $d_H(A,E) = l$*, Eve's profile* E *is uniformly distributed over* $\{X : d_H(A,X) = l\}$. The security proof leading to (2) shows that without using the additional knowledge of $(h(a_1), \ldots, h(a_n))$ Eve obtains information about K if $d_H(A,E) < d - 1$.

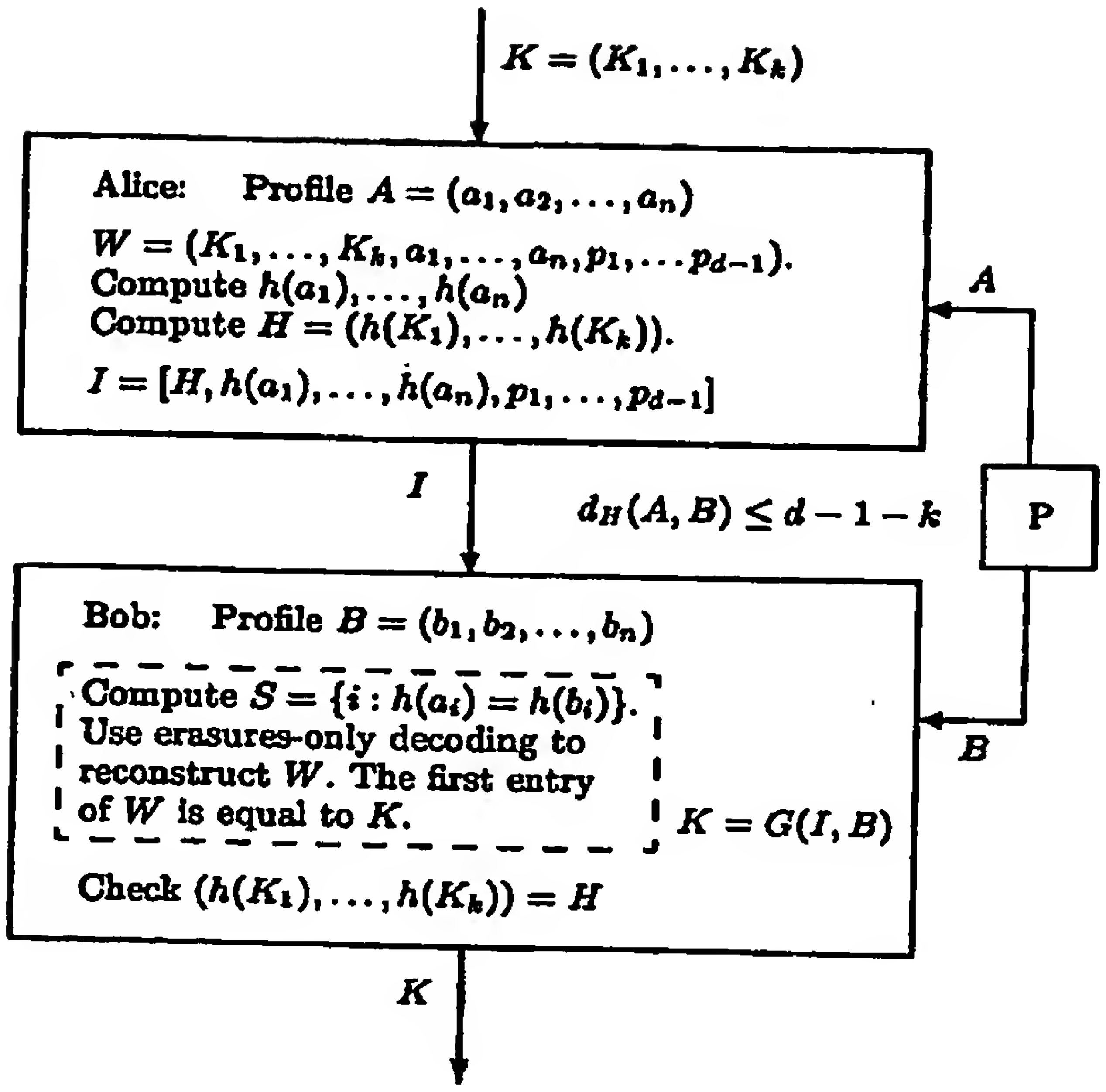
$K = (K_1,\dots,K_k)$
Alice: Profile $A = (a_1,a_2,\dots,a_n)$
$W = (K_1,\dots,K_k,a_1,\dots,a_n,p_1,\dots p_{d-1})$.
Compute $h(a_1),\dots,h(a_n)$
Compute $H = (h(K_1),\dots,h(K_k))$.
$I = [H,h(a_1),\dots,h(a_n),p_1,\dots,p_{d-1}]$
A
I
$d_H(A,B) \leq d-1-k$
P
Bob: Profile $B = (b_1,b_2,\dots,b_n)$
Compute $S = \{i : h(a_i) = h(b_i)\}$.
Use erasures-only decoding to reconstruct W. The first entry of W is equal to K.
$K = G(I,B)$
B
Check $(h(K_1),\dots,h(K_k)) = H$
K

Fig. 3. H-Protocol: Profile matching based on a one-way function $h(.)$.

Let $d_H(A,E) = d-1$. Without loss of generality, let the first $l = n-d+1$ entries of E be equal to the first l entries of A. Let $K' = (K'_1,\dots,K'_k)$. Then there exists a unique code word

$$W' = (K'_1,\dots,K'_k,a_1,\dots,a_l,a'_{l+1},\dots,a'_n,p_1,\dots,p_{d-1})$$

because the positions outside $a'_{l+1},\dots,a'_n$ form an information set (each set of size $k+l+d-1 = k+n$ is an information set). This defines the mapping (linear over $GF(q)$)

$$\phi(K') = (a'_{l+1},\dots,a'_n).$$

Notice that

$$\left.\begin{array}{l}(h(a_{l+1}),\dots,h(a_n)) \neq (h(a'_{l+1}),\dots,h(a'_n)),\\ \phi(K') = (a'_{l+1},\dots,a'_n)\end{array}\right\} \Rightarrow K \neq K'.$$

By our assumption on the distribution of E, the mapping $\phi(.)$ represents the information about K given Eve's knowledge given by her profile E, the set of positions $\{i : e_i = a_i\}$, and $(h(a_{l+1}),\dots,h(a_n))$.

Suppose that Eve has the computational power to perform up to $M \geq 1$ evaluations of the one-way function $h(.)$. We distinguish two cases. In the first case, we assume that Eve is not lucky and only computes $h(a_i')$'s for $a_i' \neq a_i$. Then Eve can compute at most

$$\left(\frac{M}{n-l}\right)^{n-l} \leq 2^{(d-1)\log_2 M}$$

tuples $(h(a_{l+1}'), \ldots, h(a_n'))$. Hence, Eve can exclude at most $2^{(d-1)\log_2 M}$ possibilities for the secret key K. The remaining possibilities for K are all equally likely. This proves that information theoretically Eve gets a negligible amount of information about K if the size of K is $> (d-1)\log_2 M$. For example, if the size of K is $1 + (d-1)\log_2 M$ bits then Eve obtains 1 bit of information.

In the second case, we assume that Eve is lucky and that she computes $h(a_i') = h(a_i)$, hence, $a_i' = a_i$. Let $i = l+1$ without loss of generality. Now the positions outside $K_k', a_{l+2}', \ldots, a_n'$ form an information set. This defines the mapping

$$\phi'(K_1', \ldots, K_{k-1}') = (K_k', a_{l+2}', \ldots, a_n').$$

In particular, $\phi'(K_1, \ldots, K_{k-1}) = (K_k, \ldots)$. In other words, Eve obtains $\log_2 q$ bits of information about K. (This also shows how Eve obtains partial information about K in the protocol based on RS codes if $d_H(A,E) < d-1$.) By our assumption on the distribution of E, the probability that Eve computes $h(a_i') = h(a_i)$ is at most M/q. This proves the following lemma.

Lemma 3. *Suppose that, given $d_H(A,E) = l$, Eve's profile E is uniformly distributed over $\{X : d_H(A,X) = l\}$. Suppose that Eve has the computational power to perform up to $M \geq 1$ evaluations of the one-way function $h(.)$, $\log_2 M < \log_2 q$. Then, in the protocol in Figure 3, there exists a distance d such that Bob can reconstruct $K = (K_1, \ldots, K_k)$ and Eve obtains, with probability at least $(1 - M/q)$, at most 1 bit of information about K if and only if $d_H(A,B) \leq d_H(A,E) - k$ and if the size of $K = (K_1, \ldots, K_k)$ is at least $1 + (d-1)\log_2 M$ bits.*

Since the security is based on the difficulty of inverting the one-way function $h(.)$, there is no need for Alice and Bob to share a key with more bits than the number of input bits of the one-way function, hence, $k = 1$. If we assume that Eve is not malicious (or over-curious) and Eve only tries the protocol of Figure 3 to obtain the secret K, then $M = 1$. In this case Lemma 3 shows that by using one-way functions we can solve the profile matching problem if $d_H(A,B) < d_H(A,E)$. This is perfect (as in perfect secret sharing schemes) in the sense that either one can reconstruct K or one does not obtain any information about K.

In Section 4.3 we generalize Lemma 3 towards arbitrary distributions.

3.3 Collusions

We may extend our model and require that multiple Eve's may collude to obtain information about K. Even though each Eve has a different profile with a large

RS-Protocol (Fig. 2):	H-Protocol (Fig. 3):
$d_H(A,B) \leq (d-1-k)/2$	$d_H(A,B) \leq d-1-k$
$d_H(A,B) \geq d-1$	$d_H(A,B) \geq d-1$
$\lvert K \rvert = k \log_2 q > \lceil \log_2 M \rceil$	$\lvert K \rvert = k \log_2 q > \lceil (d-1) \log_2 M \rceil$

Table 1. Restrictions on the parameters defining the protocol based on RS codes and the protocol using one-way functions.

distance to A, together they may obtain a profile with small distance to A. In the H-Protocol of Figure 3, by comparing $h(a_i)$ and $h(e_i)$, each Eve finds out which of the entries in her profile are equal to the corresponding entries in A. As soon as a collusion of Eve's obtains enough matching entries then they can reconstruct K.

The RS-Protocol of Figure 2 offers slightly more security because only the parities of the code word W are revealed. However, by using list decoding [GS99] and by comparing the lists of most likely code words of each Eve, the collusion of Eve's may obtain a significant amount of information about K.

Notice that in the H-Protocol of Figure 3 Alice may partly obfuscate the vector $(h(a_1), \ldots, h(a_n))$ by using the technique of Figure 2. Alice encodes $(h(a_1), \ldots, h(a_n))$ into a

$$[n + 2(d-1-k), n, 2(d-1-k)+1]$$

RS code word and transmits instead of $(h(a_1), \ldots, h(a_n))$ the parities of the corresponding codeword to Bob. If $d_H(A,B) \leq d-1-k$, see (3), then Bob can reconstruct this RS code word, in particular the vector $(h(a_1), \ldots, h(a_n))$. Since $d_H(A,E) \geq d-1$, see (2), Eve does not obtain complete knowledge about $(h(a_1), \ldots, h(a_n))$.

3.4 Public Feedback Channel

If we allow Bob to communicate to Alice over the public channel, then a more straightforward solution is possible. Alice computes

$$I = [h(a_1), \ldots, h(a_n)],$$

which she transmits to Bob. Bob computes the set

$$S = \{i : h(a_i) = h(b_i)\} = \{i : a_i = b_i\},$$

which he transmits to Alice. If Alice wants to share a key with anyone who knows a_i for $i \in S$, then Alice transmits

$$K + \mathrm{Hash}(a_i : i \in S)$$

to Bob. Bob computes $\texttt{Hash}(b_i : i \in S) = \texttt{Hash}(a_i : i \in S)$ and recovers K. To avoid the last communication from Alice to Bob, Bob may choose the shared key K and transmit $K + \texttt{Hash}(a_i : i \in S)$ together with S to Alice, who will then recover K.

Eve can reconstruct K if $h(e_i) = h(a_i)$ for $i \in S$. If one of these do not match then Eve has incomplete information about K. The amount of Eve's information about K depends on the compression factor of Hash, Eve's computational power M, and the statistics of the profiles generated by the physical system. In Section 5 we give examples.

Notice that the proposed protocol is also a solution for profiles represented by sets instead of vectors (which represent ordered sets). Privacy-protected matching [JS02] and personal entropy systems [EHM+00] are applications. Alice first represents her profile by an arbitrary vector $(a_1, \ldots, a_n)$. She transmits the corresponding I to Bob, who computes the set

$$S = \{i : \exists_{b \in B} [h(a_i) = h(b)]\}.$$

Let $b_i \in B$ such that $h(a_i) = h(b_i)$. Then $S = \{i : a_i = b_i\}$ and the protocol proceeds as before.

Obviously the disadvantage of the new protocol is interaction between Alice and Bob. On the other hand, the advantages are

- no RS encoding or decoding,
- there are no parities to leak information about K to Eve,
- Alice does not reveal her complete profile to Bob (only the profile Bob has in common with Alice is obtained by Bob),
- Alice is in full control whether the profile of Bob has enough in common with Alice's profile (represented by set S) to share a key, and
- the profiles can be unordered sets instead of vectors.

Juels and Sudan [JS02] gave a solution for unordered sets without using a feedback channel from Bob to Alice. To ensure security they require what they call post randomization. In their solution Alice computes points $(a_i, p(a_i))$ where $p(.)$ is a polynomial with $K = p(0)$. Alice transmits these points in random order interleaved with a lot of random points (x_i, y_i) (with the x_i's distinct and unequal to any of the a_i's). The random points (in the order of 10^4 for profiles of size 22) represent the post randomization. These are needed to keep Eve uncertain about the polynomial $p(.)$. Bob is able to reconstruct $p(.)$ by using an errors-and-erasures RS decoding algorithm, see [JS02] for details. Although no feedback channel from Alice to Bob is needed, the post randomization is a practical disadvantage.

4 Unconditional Security

In the current literature the problem is investigated how Alice and Bob can generate a secret key over a noisy channel in an unconditionally secure way.

That is, the security of the key does not rely on the amount of computing time and resources that are available when attempting to obtain information about the secret key by unauthorized means. In particular, we are not allowed to use a one-way function.

As an example we assume that the physical system gives

- Alice a uniformly distributed binary vector A,
- Bob a binary vector $B = A + N_{AB}$, and
- an adversary Eve a binary vector $E = A + N_{AB} + N_{BE}$.

The physical system generates the noise vector N_{AB} with bit error probability p and the noise vector N_{BE} with bit error probability q. This situation corresponds to Wyner's wire-tap channel in which all channels are binary symmetric. If Alice wants to transmit X over the corresponding wire-tap channel, then she sends the public message $X + A$ to Bob (and Eve). Bob computes $Y = (X + A) + (A + N_{AB}) = X + N_{AB}$ and Eve computes $Z = (X + A) + (A + N_{AB} + N_{BE}) = X + N_{AB} + N_{BE}$. Without losing information about X, both Bob and Eve may discard the previous messages and keep Y and Z respectively[8].

4.1 The Broadcast Channel with Confidential Messages

Wyner's wire-tap channel is a special case of the broadcast channel with confidential messages (BCC). The BCC was introduced by Csiszár and Körner [CK78] and generalizes earlier models by Wyner [W75] and Körner and Marton [KM77]. It involves three participants: two legitimate users Alice (X) and Bob (Y), and an enemy cryptanalyst Eve (Z). Alice can communicate to Bob by using a discrete memoryless channel (DMC). It also produces side information to the enemy cryptanalyst Eve. We denote this channel by $X \to (Y, Z)$. It is defined by the transition probabilities $P_{Y,Z|X}$. Channel $X \to (Y, Z)$ and the derived channels $X \to Y$ ($P_{Y|X}$) and $X \to Z$ ($P_{Z|X}$) from Alice to Bob and from Alice to Eve are discrete and memoryless. The random variables X, Y, and Z are assumed to take values in finite sets.

In order to generate a shared secret key, Alice encodes k source symbols K^k into X^n, which is the input to the DMC $X \to (Y, Z)$. Bob produces an estimate $\hat{K}^k$ based on Y^n, the output of the channel from Alice to Bob. The block error probability is defined as

$$P_B = P(\hat{K}^k \neq K^k).$$

The coding situation can be characterized by a pair (R, Δ), where R is the *rate* at which information is sent by Alice to Bob, i.e.

$$R = H(K^k)/n,$$

[8] For example, $H(X|X + A, A + N_{AB} + N_{BE}) = H(X|X + A, X - (N_{AB} + N_{BE})) = H(X|X - (N_{AB} + N_{BE})) = H(X|Z)$ because A is uniformly distributed and statistically independent of X and $N_{AB} + N_{BE}$.

and where Δ is the enemy's per bit *equivocation* about this information, i.e.

$$\Delta = H(K^k|Z^n)/H(K^k).$$

The enemy's per bit equivocation Δ is the *fraction* of the total information that Alice seeks to transmit to Bob which remains secret to Eve.

Of course, Alice and Bob want P_B to be small, while keeping R and Δ as large as possible. The rate-equivocation pair (r, δ), $r \geq 0, \delta \geq 0$, is said to be *achievable* if, for all $\varepsilon > 0$, there exists an encoder-decoder pair such that Alice and Bob can use this pair to generate a secret key at rate

$$R \geq r - \varepsilon$$

while

$$\Delta \geq \delta - \varepsilon$$

and

$$P_B \leq \varepsilon.$$

The *capacity region* is the set of all achievable rate-equivocation pairs. The *secrecy capacity* C_s is the supremum of all information rates at which Alice and Bob can generate a key that remains essentially entirely secret for Eve, i.e., it equals the supremum of information rates r such that $(r, 1)$ is achievable.

Csiszár and Körner [CK78] characterized the capacity region in terms of information theoretical expressions. In Wyner's model X, Y, and Z form a Markov chain $X \to Y \to Z$, that is $P_{Y,Z|X} = P_{Y|X}P_{Z|Y}$. Wyner proved that

$$C_s = \max_{P_X} I(X;Y|Z) \text{ for } P_{Y,Z|X} = P_{Y|X}P_{Z|Y}. \qquad (4)$$

Massey [M83] gave a simplified treatment of Wyner's wire-tap channel. Piret [P80] showed that for Wyner's wire-tap channel C_s can be achieved by using binary linear codes in the case where $X \to Y$ and $Y \to Z$ are binary symmetric channels (he shows there exist binary linear codes without giving an explicit construction).

4.2 The BCC with Public Discussion

We can generalize the BCC by allowing public communication by Alice and Bob. The concept of the *BCC with public discussion* was introduced by Maurer in [M93] and by Ahlswede and Csiszár in [AC93]. In the BCC with public discussion the equivocation is redefined as

$$\Delta = H(K^k|Z^n, I)/H(K^k),$$

where I represents the public communication between Alice and Bob. We notice that in the definition of the information rate public communication plays no role. This is because public transmissions are assumed to be very cheap compared to transmissions over the channel $X \to (Y, Z)$. The new definition leads to the *secrecy capacity with public discussion* $\bar{C}_s$. Clearly, $C_s \leq \bar{C}_s$ which is in turn less than or equal to the capacity of the channel $X \rightarrow Y$.

Ahlswede and Csiszár found characterizations of the secrecy capacity in the restricted situation where only one public message is allowed to contain data. In the case where this public message is sent from Alice to Bob the secrecy capacity is called the *forward key-capacity* and it appears to be equal to C_s. Intuitively, this is clear since Alice can not use any information sent by Bob in order to construct her public message, simply because there are no messages sent by Bob to Alice. Hence, the public message I contains only information about the selection of the n source bits X^n. This information is transmitted to Bob and more important also to Eve. Hence, Alice and Bob are not allowed to use the part of information in X^n respectively Y^n which is dependent on I to extract a secret key. If they break this rule the secret key will depend on I, and Eve will obtain information about it. We conclude that the public message has only a negative effect on Alice's and Bob's situation. Not using a public message is better. This proves intuitively that the forward key capacity equals C_s. In the case where the non-empty public message is sent from Bob to Alice the secrecy capacity is called the *backward key-capacity*, and its characterization is unknown. In our example we deal with the forward-key capacity.

For completeness, both Maurer [M93] and Ahlswede and Csiszár [AC93] found that $\bar{C}_s$ is upper bounded by $I(X;Y|Z)$ maximized over all possible probability distributions P_X. If X, Y, and Z form a Markov chain in some order then equality holds. Thus equality holds if $X \rightarrow Y \rightarrow Z$, i.e. $P_{Y,Z|X} = P_{Y|X}P_{Z|Y}$, or $Z \leftarrow X \rightarrow Y$, i.e. $P_{Y,Z|X} = P_{Y|X}P_{Z|X}$, or $X \rightarrow Z \rightarrow Y$, i.e. $P_{Y,Z|X} = P_{Y|Z}P_{Z|X}$ (in the last case $I(X;Y|Z) = 0$). In particular, for Wyner's wire-tap channel (4),

$$\bar{C}_s = C_s = \max_{P_X} I(X;Y|Z).$$

Ahlswede and Csiszár noted that $\bar{C}_s(P_{Y,Z|X})$ is more generally upper bounded by $I(X;Y|U)$ maximized over all possible probability distributions P_X and $P_{U|Z}$ where U is some random variable;

$$\bar{C}_s \leq \max_{X,Z \rightarrow U} I(X;Y|U).$$

Alternative definitions of the secrecy capacity with public discussion are introduced and discussed in [W99].

In the situation of the BCC with public discussion only upper bounds on $\bar{C}_s$ and no precise characterizations of $\bar{C}_s$ are known. We refer to [C97,vD97,LvTvD03] for techniques about how to use public communication (the constructions require a lot of public interaction between Alice and Bob).

Summarizing, in our example (Wyner's wire-tap channel with binary symmetric channels)

$$\bar{C}_s = h_2(p(1-q) + (1-p)q) - h_2(p), \qquad (5)$$

where $h_2(x) = -x \log_2 x - (1-x) \log_2(1-x)$ denotes the binary entropy function, and

$$\bar{C}_s = C_s,$$

which means that public communication between Alice and Bob, in particular, from Alice to Bob, does not help.

4.3 One-Way Functions

If Alice and Bob are allowed to base security on the assumption that Eve has finite computing power, then they may use the following strategy. Alice transmits random vectors X_i of m bits each over Wyner's wire-tap channel to Bob. Bob receives the vectors Y_i with bit error probability p. Hence, the probability that $X_i = Y_i$ is equal to

$$(1-p)^m. \tag{6}$$

Eve receives vectors Z_i, each bit of Z_i differs from the corresponding bit in Y_i with bit error probability q. Alice transmits $h(X_i)$ over the public channel. As soon as Bob receives a vector Y_i such that $h(X_i) = h(Y_i)$, Alice and Bob agree on the secret key $K = X_i = Y_i$.

As soon as Alice and Bob agree on a secret key $K = X_i = Y_i$, Eve's knowledge about K is given by Z_i, $h(X_i)$, and the information that X_i and Y_i are equal to one another. If we assume that $h(X_i)$ does not reveal any information about $Y_i = X_i$, then we may measure the security by the uncertainty of Y_i given Z_i, which is equal to $h_2(q)m$. That is, given $Z_i = X_i + N_{AB,i} + N_{BE,i}$, Eve will perform an exhaustive search among the most likely candidates for Bob's subvector $Y_i = X_i + N_{AB,i}$. That is, she needs to guess $N_{BE,i}$. The most likely candidates are the ones with lowest Hamming weight. On average the Hamming weight of $N_{BE,i}$ is equal to qm. This means that on average Eve needs to perform an exhaustive search among all binary vectors of length m in the sphere around the all-zero vector and with radius qm. In this sense a security of s' bits means that the number of vectors in the sphere

$$\sum_{j=0}^{qm} \binom{m}{j} \approx 2^{h_2(q)m}$$

should be equal to $2^{s'}$. We conclude that

$$s' = h_2(q)m. \tag{7}$$

By using the next lemma we will show that the definition of security is more subtle.

Lemma 4. *Let Y and Z be random variables in $\mathcal{Y}$ and $\mathcal{Z}$. Let U be a set which consists of M elements in $\mathcal{Y}$ and define*

$$\delta_U(y) = \begin{cases} y & \text{for } y \in U, \\ ? & \text{for } y \notin U. \end{cases}$$

Then,

$$H(Y|Z = z, \delta_U(Y)) \geq H(Y|Z = z) - \varepsilon \log_2 M - h_2(\varepsilon),$$

where

$$\varepsilon = P(\delta_U(Y) \neq ?|Z = z).$$

Proof. The proof is similar to the proof of Fano's inequality. We derive

$$\begin{aligned} H(Y|Z=z) &= H(Y,\delta_U(Y)|Z=z) \\ &= H(\delta_U(Y)|Z=z) + H(Y|Z=z,\delta_U(Y)). \end{aligned} \tag{8}$$

Define random variable δ to be equal to 1 if $\delta_U(Y) \neq ?$ and equal to 0 if $\delta_U(Y) = ?$. Then

$$\begin{aligned} H(\delta_U(Y)|Z=z) &= H(\delta_U(Y),\delta|Z=z) \\ &= H(\delta|Z=z) + H(\delta_U(Y)|Z=z,\delta), \end{aligned} \tag{9}$$

where

$$H(\delta|Z=z) = h_2(P(\delta_U(Y) \neq ?|Z=z)) = h_2(\varepsilon) \tag{10}$$

and

$$\begin{aligned} H(\delta_U(Y)|Z=z,\delta) = P(\delta=0|Z=z)H(\delta_U(Y)|Z=z,\delta=0) + \\ P(\delta=1|Z=z)H(\delta_U(Y)|Z=z,\delta=1). \end{aligned} \tag{11}$$

Notice that

$$H(\delta_U(Y)|Z=z,\delta=0) = H(\delta_U(Y)|Z=z,\delta_U(Y)=?) = 0, \tag{12}$$

$$P(\delta=1|Z=z) = P(\delta_U(Y) \neq ?|Z=z) = \varepsilon, \tag{13}$$

and

$$\begin{aligned} H(\delta_U(Y)|Z=z,\delta=1) &= H(\delta_U(Y)|Z=z,\delta_U(Y) \neq ?) \\ &\leq \log_2 |U| = \log_2 M. \end{aligned} \tag{14}$$

The combination of (8-14) proves the lemma.

□

Corollary 1. *Let Y and Z be random variables which take values in $\{0,1\}^m$ such that Y is uniformly distributed and*

$$P_{Z|Y}(z|y) = q^{d_H(z,y)}(1-q)^{m-d_H(z,y)}, \quad q \leq 1/2.$$

Let U be a set which consists of $\binom{m}{um} \approx 2^{h_2(u)m}$, $u < q$, binary vectors of length m and define

$$\delta_U(y) = \begin{cases} v & \text{for } y \in U, \\ ? & \text{for } y \notin U. \end{cases}$$

Then,

$$P(\delta_U(Y) \neq ?|Z=z) \leq \varepsilon = \sum_{j=0}^{um} \binom{m}{j} q^j (1-q)^{m-j}$$

and

$$H(Y|Z=z,\delta_U(Y)) \geq (h_2(q) - \varepsilon h_2(u))m - h_2(\varepsilon).$$

Proof. The probability $P(\delta_U(Y) \neq ?|Z = z)$ is maximized by taking U to be the sphere around z with radius um. Since $h_2(x)$ is monotone increasing for $0 \leq x \leq 1/2$, the corollary follows from Lemma 4.

□

Let us apply Corollary 1 to the situation of Eve. Suppose that Eve's computing resources allow her to perform at most $M = 2^{h_2(u)m}$, $u < q$, evaluations of a one-way function $h(.)$. Then she is able to construct a set U of size M for which Eve knows $\delta_U(Y_i)$. This means that if $Y_i \in U$ then Eve has full knowledge of $Y_i = X_i = K$ and if $Y_i \notin U$ then Eve has $H(Y_i|Z_i = z_i, \delta_U(Y_i) = ?)$ uncertainty about $Y_i = X_i = K$. Thus Eve's average uncertainty about K is equal to

$$P(\delta_U(Y_i) = ?|Z_i = z_i)H(Y_i|Z_i = z_i, \delta_U(Y_i) = ?) = H(Y_i|Z_i = z_i, \delta_U(Y_i)).$$

According to Corollary 1 this is at least

$$s = (h_2(q) - \varepsilon h_2(u))m - h_2(\varepsilon). \tag{15}$$

where

$$\varepsilon = \sum_{j=0}^{um} \binom{m}{j} q^j (1-q)^{m-j}.$$

So, the security s is measured by (15) and not by (7). Notice that for $q \approx u$ we obtain $\varepsilon \approx 1$ and $s \approx 0$. This corresponds with our intuition; if Eve is able to construct a set U of typical sequences y given $Z_i = z$ then $Y_i \in U$ with probability close to one and Eve is able to reconstruct $Y_i = X_i = K$.

Since K only contains s bits of security, Alice and Bob should compress K to s bits. This is called privacy amplification [C97]. In general, multiplication with a random binary $m \times s$ matrix leads to a key of s bits about which Eve has only negligible information [CH77]. Of course Alice and Bob can also use a hash function [MvOV96].

Combination of (6) and (15) gives the secrecy rate of Alice's and Bob's strategy;

$$(1-p)^m s$$

bits per vector of m bits. See (5), the secrecy capacity with public discussion equals

$$\bar{C}_s = m(h_2(p + q - 2pq) - h_2(p))$$

bits per vector of m bits. In this example, the secrecy rate of Alice and Bob does not generally improve[9] $\bar{C}_s$. Notice that Alice and Bob's secrecy rate gets better if Bob has computational power to evaluate $h(.)$ and construct a set U for himself.

[9] We conjecture that it only improves if Eve has no computing power at all, that is $M = 0$. We checked that there is no improvement for $q = 1/2$ or for $p < u$ with $m = (\log_2 M)/h_2(u)$. If $M = 0$ then $s = mh_2(q)$ and for small p and $m \ll 1/p$ we obtain $(1-p)^m s = (1 - mp + O((pm)^2))mh_2(q)$ and $C_s = m(h_2(q) + p(1-2q)\log_2 \frac{1-q}{q} - p\log_2 p + O(p^2))$. Hence, $(1-p)^m s - C_s = pm(-\log_2 p - (1-2q)\log_2 \frac{1-q}{q} - mh_2(q) + O(pm^2))$ which is positive for $m \ll -\log_2 p$ and p small enough.

The technique with one-way functions shows that for Wyner's wire-tap channel with binary symmetric channels a simple strategy is possible and that we can prove its security (see Section 5.2 for a numerical example). It remains unsolved when simple strategies involving one-way functions improve the secrecy capacity with public discussion.

Notice that the BCC with public discussion in which computational security is allowed makes no sense. If cryptography primitives are allowed then the public channel itself can be used to share a secret key, see for example Shamir's no-key protocol [MvOV96]. In the BCC only Alice can transmit messages to Bob over the noisy channel. Therefore Alice can authenticate herself by using the noisy channel to transmit a sensible message encoded and encrypted with the secret key. The difference with our model is that anyone (not just Bob) who has a profile close enough to Alice's profile should be able to learn the key.

As a final remark we notice that Lemma 4 can be used to generalize Lemma 3 in Section 3.2.

Lemma 5. *Let (A_i, E_i), $1 \leq i \leq n$, be independently distributed pairs of random variables. Suppose that Eve has the computational power to perform up to $M \geq 1$ evaluations of the one-way function $h(.)$, $\log_2 M < H(A_i)$. Then, in the protocol in Figure 3, Eve obtains at most*

$$I(A;E) + H(P) + \frac{M \log_2 M}{\min_i 2^{H(A_i)}} + 1$$

bits of information about KA.

Proof. Let $Y = A_i$ and let $Z = E_i$ in Lemma 4. Eve is able to construct a set U_i of size $M_i \leq M$ for which Eve knows $\delta_{U_i}(A_i)$. By Lemma 4,

$$H(A_i|E_i = z, \delta_{U_i}(A_i)) \geq H(A_i|E_i = z) - \varepsilon_i \log_2 M_i - h_2(\varepsilon_i),$$

where

$$\varepsilon_i = P(\delta_{U_i}(A_i) \neq ?|E_i = z).$$

This proves (notice that $h_2(x)$ is $\cap$-convex)

$$H(A_i|E_i, \delta_{U_i}(A_i)) \geq H(A_i|E_i) - \varepsilon_i' \log_2 M_i - h_2(\varepsilon_i'),$$

where

$$\varepsilon_i' = P(\delta_{U_i}(A_i) \neq ?) = P(A_i \in U_i).$$

By using the concept of typical sets we observe that

$$\varepsilon_i' = P(A_i \in U_i) \leq M/2^{H(A_i)}.$$

Let $\delta_U(A) = (\delta_{U_i}(A_1), \ldots, \delta_{U_i}(A_n))$. Then Lemma 2 with E replaced by $E\delta_U(A)$ gives

$$\begin{aligned}
I(KA; E\delta_U(A)P) &\leq I(A; E\delta_U(A)) + H(P) \\
&= H(P) + \sum_{i=1}^{n} I(A_i; E_i\delta_{U_i}(A_i)) \\
&= H(P) + \sum_{i=1}^{n} (H(A_i) - H(A_i|E_i\delta_{U_i}(A_i))) \\
&\leq H(P) + \sum_{i=1}^{n} (H(A_i) - H(A_i|E_i) + \varepsilon'_i \log_2 M + h_2(\varepsilon'_i)) \\
&= I(A; E) + H(P) + \sum_{i=1}^{n} \left(\frac{M_i \log_2 M_i}{2^{H(A_i)}} + h_2(M_i/2^{H(A_i)}) \right) \\
&\leq I(A; E) + H(P) + \frac{M \log_2 M}{\min_i 2^{H(A_i)}} + h_2(M/\min_i 2^{H(A_i)}) \\
&\leq I(A; E) + H(P) + \frac{M \log_2 M}{\min_i 2^{H(A_i)}} + 1
\end{aligned}$$

□

This lemma can also be used in Section 3.4 to determine the compression factor of the hash function Hash.

5 Examples

5.1 Hashed Binary Symmetric Channel

Let us consider the RS-Protocol of Figure 2 and the H-Protocol of Figure 3, where we allow computational security. Suppose that Alice, Bob, and Eve have n subvectors of length m each, that is

$$X = (X_1, \ldots, X_n), \; Y = (Y_1, \ldots, Y_n), \text{ and } Z = (Z_1, \ldots, Z_n),$$

with each X_i, Y_i, and Z_i containing m bits. Alice uses RS encoding (by representing each X_i as a symbol in $GF(2^m)$) to compute the RS code word

$$W = [K_1, \ldots, K_k, X_1, \ldots, X_n, p_1, \ldots, p_{d-1}].$$

Alice transmits

$$I^{RS} = [h(K_1, \ldots, K_k), p_1, \ldots, p_{d-1}]$$

or

$$I^{H} = [h(K_1), \ldots, h(K_k), h(X_1), \ldots, h(X_n), p_1, \ldots, p_{d-1}]$$

over the public channel to Bob depending on which protocol she uses. In the protocol using one-way functions Bob computes $h(Y_i)$. Only if $h(X_i) = h(Y_i)$

then Alice and Bob know that with overwhelming probability $X_i = Y_i$ (that is the corresponding entries in N_{AB} are zero).

We assume that X is uniformly distributed and that, given $d_H(Z, X) = l$, Z is uniformly distributed over

$$\{Z : d_H(Z, X) = |\{i : Z_i \neq X_i\}| = l\}$$

(this assumption is used in the derivation of (2)). For example,

$$P_{Z_i|X_i}(z|x) = \begin{cases} 1-q, & \text{if } z = x, \\ q/(2^m - 1), & \text{if } z \neq x, \end{cases}$$

and similarly

$$P_{Y_i|X_i}(y|x) = \begin{cases} 1-p, & \text{if } y = x, \\ p/(2^m - 1), & \text{if } y \neq x. \end{cases}$$

Since the probability that $h(X_i) = h(Z_i)$ is equal to $1-q$, Eve gets to know at least

$$t_E = n - d + 2 \tag{16}$$

subvectors X_i (such that $d(X, Z) \leq d-2$) with which she can obtain information about K (see (2)) with probability

$$P_E = \sum_{j=t_E}^{n} \binom{n}{j} (1-q)^j q^{n-j} \approx q^n (n(1-q))^{t_E} / t_E! \tag{17}$$

where the approximation holds for

$$(1-q)n < t_E. \tag{18}$$

The probability that Bob gets to know at least

$$t_B^{RS} = n - (d-1-k)/2 \text{ resp. } t_B^H = n - (d-1-k) \tag{19}$$

subvectors X_i (such that $d_H(X, Y)$ is small enough) with which he can recover K, is equal to

$$1 - P_{AB} = 1 - \sum_{j=0}^{t_B^*-1} \binom{n}{j} (1-p)^j p^{n-j} \approx 1 - p^n (n(1-p))^{t_B^*-1} / (t_B^* - 1)!, \tag{20}$$

where the approximation holds for

$$n(1-p) > t_B^*. \tag{21}$$

Notice that conditions (18) and (21) are necessary to achieve small P_E and P_{AB}.

As an example, suppose that the physical system $\mathbf{P}$ in the profile matching model generates binary vectors Z_i' and Y_i' of length m as the outputs of two binary symmetric channels with random input X_i' characterized by bit error probabilities 0.01 and 0.05 respectively. Suppose that the physical system uses a one-way function to compute $X_i = h(X_i')$, $Y_i = h(Y_i')$, and $Z_i = h(Z_i')$. Then we obtain the situation described in this section, where

$$p = 1 - (1 - 0.01)^m = 1 - 0.99^m$$

and

$$q = 1 - (1 - 0.05)^m = 1 - 0.95^m.$$

Let

$$n \approx \alpha t_B^* / 0.99^m \approx \beta t_E / 0.95^m \tag{22}$$

with $\alpha > 1$ and $\beta < 1$ such that the conditions (18) and (21) are satisfied. Notice that $(1-x)^{1/x} \leq e^{-1}$, for $0 < x \leq 1$, and $x^x/x! \leq e^x$, for $x \geq 0$. By using the approximations (17) and (20) we derive

$$\begin{aligned} P_E &\approx (1 - 0.95^m)^n (n \cdot 0.95^m)^{t_E} / t_E! \\ &\leq e^{-\beta t_E} (\beta t_E)^{t_E} / t_E! \\ &\leq (e^{1-\beta} \beta)^{t_E} \end{aligned} \tag{23}$$

and

$$\begin{aligned} P_{AB} &\approx (1 - 0.99^m)^n (n \cdot 0.99^m)^{t_B^*} / t_B^*! \\ &\leq e^{-\alpha t_B^*} (\alpha t_B^*)^{t_B^*} / t_B^*! \\ &\leq (e^{-(\alpha - 1)} \alpha)^{t_B^*}. \end{aligned} \tag{24}$$

See equations (16) and (22), $t_E = n - d + 2 = \beta t_E / 0.95^m - d + 2$, hence,

$$d - 2 = (\beta / 0.95^m - 1) t_E. \tag{25}$$

Since $d - 2$ is positive for $d \geq 2$, this leads to the condition

$$m \geq \frac{\ln \beta}{\ln 0.95} = -19.50 \cdot \ln \beta. \tag{26}$$

See equation (19),

$$t_B^{RS} = n - (d - 1 - k)/2 = \alpha t_B^{RS} / 0.99^m - (d - 1 - k)/2.$$

Combined with (25) this gives

$$(\beta / 0.95^m - 1) t_E + 1 - k = d - 1 - k = 2(\alpha / 0.99^m - 1) t_B^{RS}.$$

Since $k \geq 1$, this leads to the condition (use (22))

$$2 \cdot 0.99^m / \alpha - 0.95^m / \beta > 1. \tag{27}$$

In particular, $0.99^m > 1/2$, hence, $m \leq 68$.

See equation (19),

$$t_B^H = n - (d - 1 - k) = \alpha t_B^H / 0.99^m - (d - 1 - k).$$

Combined with (25) this gives

$$(\beta/0.95^m - 1)t_E + 1 - k = d - 1 - k = (\alpha/0.99^m - 1)t_B^H.$$

In this case we also have the condition $km > (d-1)\log_2 M$ with $\log_2 M < m$, that is

$$k > (d-1)(\log_2 M)/m = ((\beta/0.95^m - 1)t_E - 1)(\log_2 M)/m.$$

This leads to the condition (use (22))

$$\frac{m}{m - \log_2 M} \cdot 0.99^m/\alpha - 0.95^m/\beta > \frac{\log_2 M}{m - \log_2 M}. \qquad (28)$$

In particular, $1 < \alpha < m0.99^m/\log_2 M$, and, since $m0.99^m$ is maximized for $m \approx 99$, we obtain $\log_2 M \leq 35$. So, *in this parameter setting Alice and Bob cannot protect themselves against an adversary with* $M > 2^{35}$. This shows that the protocol which uses one-way functions has its limitations.

Let us continue with the RS based protocol and fix $m = 29$ (giving $0.75 \approx 0.99^m$).Then condition (27) is satisfied for $\beta = 0.8$ and $\alpha = 1.1$ (we check (26)). We want Eve's probability of obtaining information about K to be very small for security purposes, say $P_E \leq 2^{-80} \approx e^{-55}$. Then (23) gives the restriction $t_E \geq 2377$ and together with (22) we obtain $n \approx \beta t_E/0.95^m \geq 8417$. We want the probability that Bob can reconstruct K to be large such that the profile matching protocol is robust, say $P_{AB} \leq 0.001 \approx e^{-7}$. Then (24) gives the restriction $t_B^{RS} \geq 2377$ and together with (22) we obtain $n \approx \alpha t_B^{RS}/0.99^m \geq 2198$. We take $n = 8417$ and together with (22) we obtain $t_E = 2377$ and $t_B^{RS} = 5718$. Equation (16) gives $d = 6042$ and equation (19) gives $k = 643$.

We check that (18) and (21) hold; $(1-q)n = 1902 < 2377 = t_E$ and $(1-p)n = 6289 > 5718 = t_B^{RS}$. By using Stirling's formula, $\ln x! \approx (x + 1/2)\ln x - x \ln 2$, we obtain

$$P_E \approx (1 - 0.95^{29})^{8417}(8417 \cdot 0.95^{29})^{2377}/2377! \approx e^{-1044}$$

and

$$P_{AB} \approx (1 - 0.99^{29})^{8417}(8417 \cdot 0.99^{29})^{5717}/5717! \approx e^{-7070}.$$

The approximations depend on how well (18) and (21) hold. For example, P_E in (18) is described by a binomial distribution with average $(1-q)n = 1902$ and standard deviation $\sqrt{q(1-q)n} = 38.4$. By using a Gaussian approximation of the binomial probability distribution (law of large numbers),

$$P_E \approx Q((2377 - 1902)/38.4) = Q(12.4) \approx \frac{e^{-12.4^2/2}}{\sqrt{2\pi} \cdot 12.4} = e^{-79.9},$$

where

$$Q(z) = \int_{x=z}^{\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}}\, \mathrm{d}x$$

and

$$(1 - 1/z^2)\frac{e^{-z^2/2}}{\sqrt{2\pi} \cdot z} < Q(z) < \frac{e^{-z^2/2}}{\sqrt{2\pi} \cdot z},$$

for $z > 0$ [W96]. Similarly,

$$P_B \approx Q((6289 - 5718)/39.9) = Q(14.3) \approx \frac{e^{-14.3^2/2}}{\sqrt{2\pi} \cdot 14.3} = e^{-105.8}.$$

This shows how the approximations (18) and (21) only lead to a first solution. We only require $P_E \approx e^{-55}$. This suggests to rescale n, d, and k with $55/79.9 = 0.688$ such that $n = 5794$, $d = 4157$, $k = 442$, with $m = 29$, $P_E \approx e^{-56.0}$, and $P_{AB} \approx e^{-73.5}$. This means that the shared key has size $km = 12818$ bits and that the physical system generates $nm = 168026$ bits in total. The secrecy rate is equal to 0.0763.

A similar example can be worked out for the protocol based on one-way functions. However, if we allow feedback our scheme becomes more simple. We choose $m = 160$, such that an adversary Eve with $M = 2^{80}$ computing power and with a profile for which $Z_i \neq X_i$ ($h(Z_i) \neq h(X_i)$) has at most a probability $M/2^m = 2^{-80} = 1/M$ of successfully guessing X_i' with $h(X_i') = h(X_i)$. This means that despite Eve's computing power, Eve can only get to know information about X_i if $h(X_i) = h(Z_i)$, that is, $X_i = Z_i$. This happens with probability $0.95^{160} = 0.000273$.

Bob receives $Y_i = X_i$ with probability $0.99^{160} = 0.200$. This shows that Bob receives at least $t = \gamma n$ matching $Y_i = X_i$ about which Eve does not obtain any information (that is, $Z_i \neq X_i$) with probability

$$1 - \sum_{j=0}^{\gamma n - 1} \binom{n}{j} w^j (1-w)^{n-j} \approx 1 - Q((w - \gamma)n/\sqrt{w(1-w)n}),$$

where

$$w = 0.200 \cdot (1 - 0.000273).$$

For $\gamma \leq w$,

$$Q((w-\gamma)n/\sqrt{w(1-w)n}) < \frac{e^{-n(w-\gamma)^2/(2w(1-w))}\sqrt{w(1-w)}}{\sqrt{2\pi} \cdot (w-\gamma)\sqrt{n}}.$$

If we want to achieve a robustness and security of $e^{-55} \approx 2^{-80}$, then $n(w - \gamma)^2/(2w(1-w)) \approx 55$ which is equivalent to $n(0.20 - \gamma)^2 \approx 17.60$ and

$$\gamma = 0.20 - \sqrt{17.60/n}.$$

Bob communicates the set of positions i with matching $Y_i = X_i$ to Alice. Both Alice and Bob compute a hash of the matching part of their profiles. The hash compresses the matching profiles to a secret key K of $tm = \gamma nm$ bits. Only with probability at most $e^{-55} \approx 2^{-80}$ Eve obtains information about K. The total number of bits in a profile is nm. Hence, the secrecy rate is equal to γ, which is close to 0.20 for large n. For $\gamma = 0.0763$ as in our previous example, $n \approx 1150$ and the total number of bits in a profile is equal to $1150 \cdot 160 = 184000$.

For $t = 1$, we can do exact computations. Bob receives at least $t = 1$ entries $Y_i = X_i$ about which Eve does not obtain any information with probability $1-(1-w)^n$. We want to achieve a robustness and security of $2^{-80} = (1-w)^n$. Taking $n = 35$ gives a solution where the total number of bits in a profile is equal to $mn = 5600$ bits and the size of K is $m = 160$ bits.

5.2 Binary Symmetric Channel

Suppose that the physical system in the profile matching model generates binary vectors $Y = (Y_1, \ldots, Y_n)$ and $Z = (Z_1, \ldots, Z_n)$ of length n as the output of binary symmetric channels with random input $X = (X_1, \ldots, X_n)$ and characterized by bit error probabilities $p = 0.01$ and $q = 0.05$ respectively. Suppose Alice and Bob use the RS based protocol of of Figure 2 to share a secret key. Let

$$m = \lceil \log_2(k + n + d - 1) \rceil \tag{29}$$

and consider a $[k+n+d-1, k+n, d]$ RS code over $GF(2^m)$. Alice uses RS encoding to compute the RS code word

$$W = [K_1, \ldots, K_k, X_1, \ldots, X_n, p_1, \ldots, p_{d-1}]$$

(notice that the bits X_i are embedded as elements in $GF(2^m)$). Alice transmits

$$I = [h(K_1, \ldots, K_k), p_1, \ldots, p_{d-1}]$$

over the public channel to Bob.

As we have seen in Section 3.1 (see (1)), Bob can reconstruct the code word W if $d_H(X, Y) \leq (d-1-k)/2$. Let

$$t_B = n - (d-1-k)/2 = (1-\gamma)n \tag{30}$$

with $1 - p > 1 - \gamma$. Then Bob can reconstruct W with probability

$$\begin{aligned} 1 - P_{AB} &= 1 - \sum_{j=0}^{t_B - 1} \binom{n}{j} (1-p)^j p^{n-j} \\ &\approx 1 - Q((\gamma - p)n/\sqrt{p(1-p)n}) \\ &\approx 1 - \frac{e^{-n(\gamma-p)^2/(2p(1-p))}\sqrt{p(1-p)}}{\sqrt{2\pi} \cdot (\gamma - p)\sqrt{n}}. \end{aligned} \tag{31}$$

From (29) and (30) we infer that

$$m = \lceil \log_2(2k + (1 + 2\gamma)n) \rceil. \tag{32}$$

Suppose that Alice and Bob want to achieve $P_{AB} \leq 0.001 \approx e^{-7}$. Then, according to (31), we need to choose

$$n(\gamma - p)^2/(2p(1-p)) \approx 7,$$

that is,

$$\gamma = 0.01 + \sqrt{0.1386/n}. \tag{33}$$

Notice that we cannot use Lemma 1 because the assumption on the distribution of Eve's profile does not hold. Instead we use Lemma 2, which says that Eve obtains at most

$$I(X;Z) + H(P) = (1 - h_2(q))n + (d-1)m$$

bits of information about KX. After Bob has reconstructed KX, both Alice and Bob use privacy amplification to distill from KX

$$\begin{aligned} & H(KX) - ((1 - h_2(q))n + (d-1)m) \\ &= (km + n) - ((1 - h_2(q))n + (d-1)m) \\ &= h_2(q)n - (d - 1 - k)m \end{aligned} \tag{34}$$

secret key bits. In combination with (32) and (33), (34) is maximized for $k = 0$ and equal to

$$(0.286 - (0.02 + 0.745/\sqrt{n})\lceil \log_2(1.02 \cdot n + 0.745 \cdot \sqrt{n})\rceil)n.$$

If we want a positive number of secret key bits then $0.286 > (0.02{+}0.745/\sqrt{n})$, hence, $n \geq 7$. For $n \geq 7$, $\lceil \log_2(1.02 \cdot n + 0.745 \cdot \sqrt{n})\rceil \geq 4$. Notice that $0.286 > (0.02 + 0.745/\sqrt{n}) \cdot 4$ for $n \geq 210$. For $n \geq 210$, $\lceil \log_2(1.02 \cdot n + 0.745 \cdot \sqrt{n})\rceil \geq 8$. Notice that $0.286 > (0.02 + 0.745/\sqrt{n}) \cdot 8$ for $n \geq 2238$. For $n \geq 2238$, $\lceil \log_2(1.02 \cdot n + 0.745 \cdot \sqrt{n})\rceil \geq 12$. Notice that $0.286 > (0.02 + 0.745/\sqrt{n}) \cdot 12$ for $n \geq 37772$. For $n \geq 37772$, $\lceil \log_2(1.02 \cdot n + 0.745 \cdot \sqrt{n})\rceil \geq 16$. There does not exist an n such that $0.286 > (0.02 + 0.745/\sqrt{n}) \cdot 16$. So, the protocol based on RS codes does not lead to a solution.

Our feedback strategy gives a simple and practical solution. We will use Corollary 1 to prove its security. Let Eve's computing power be represented by $2^{80} = M = 2^{h_2(u)m}$ with $u < q$. For example, $u = q/2 = 0.025$ with $h_2(u) = 0.169$ and $m = 80/h_2(u) = 474$. We derive

$$\varepsilon \approx Q((q-u)\sqrt{m}/\sqrt{q(1-q)}) = Q(3.486) \leq e^{-3.486^2/2}/\sqrt{2\pi} \cdot 3.486 = 0.000263.$$

Hence, Corollary 1 states that Eve's uncertainty about $(X_1, \ldots, X_m)$ given her knowledge of $h(X_1, \ldots, X_m)$ and $(Z_1, \ldots, Z_m)$ is approximately equal to $h_2(q)m = 0.286 \cdot 474 = 135$ bits. This means that Alice and Bob can use privacy amplification to distill 135 secret bits.

The probability that Bob receives a string of $m = 474$ bits $Y_i = X_i$ is equal to $0.99^m = 0.00853$. Suppose that Bob receives n strings of m bits. Then the probability that none of the strings matches with the corresponding strings received by Alice is equal to $(1 - 0.00853)^n$. If we want this probability to be ≤ 0.001, we need to take $n = 807$. Summarizing, Alice and Bob need $mn = 382518$ bits to extract a key of 135 bits.

6 Applications

6.1 Biometrics

Figure 4 depicts the model for biometrics as introduced in [LT03,VTD+03]. The physical system P measures Alice's fingerprint X. In the set-up phase Alice uses P to measure her fingerprint X. Like in Figure 3, the resulting measurement A is used to compute $I = [H, h_A, p_A]$. The first part of I consisting of $H = h(K)$ and h_A are already given to Bob who stores these values in a database. Since the database only contains images of one-way functions, the security is not compromised if the database is publicly accessible.

Bob's ATM machine may measure Alice's fingerprint X a second time. This gives Bob the measured fingerprint B. Since A and B are different measurements at different times using possibly different measuring devices, A and B are in general not equal to one another. Since A and B are measurements of the same fingerprint X, $d(A, B)$ is small. This means that Bob can use our solution to reconstruct K and check its commitment $H = h(K)$.

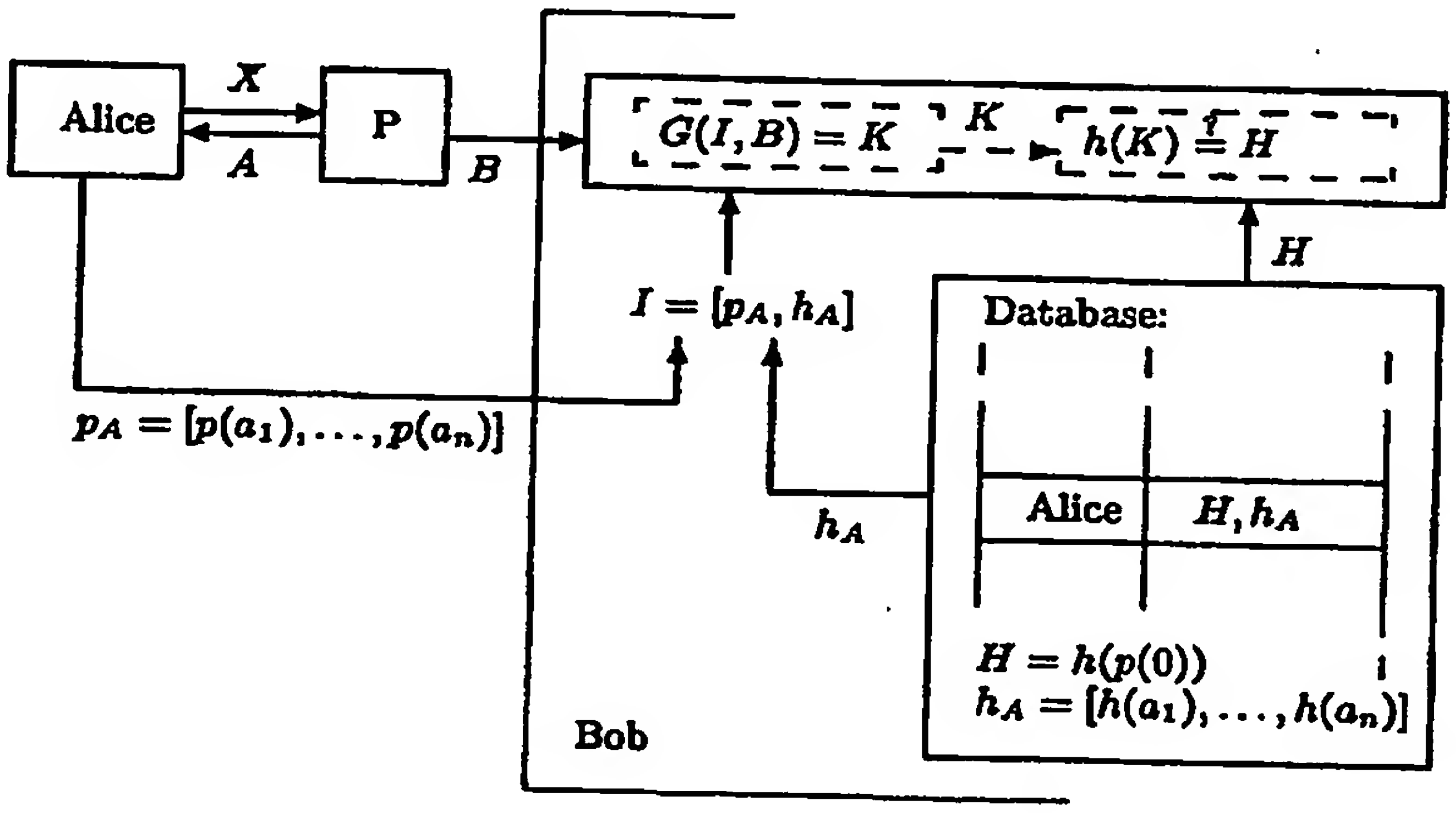

Fig. 4. Model for biometrics.

In biometrics the role of Eve is of an adversary who tries to obtain copies from Alice's fingerprint by using for example a disposed coffee cup which contains Alice's fingerprint.

Notice that at any time Alice may use A to compute a new p_A and a new commitment H corresponding to a new K. If Alice transmits to Bob the new p_A with the new commitment H, then Bob can update its database and Alice and Bob can start using the new K.

Figure 4 corresponds to the model presented in [LT03,VTD+03] in that function $G(I, B)$ (see Figure 3 for its definition) is

- *contracting*, that is for any arbitrary given value of K and any arbitrary profiles A and B with $d(A, B)$ small enough, its is not computationally difficult to find at least one value I such that

$$G(I, A) = G(I, B) = K,$$

and

- *revealing*, that is for any A it is not computationally difficult to find a value of I such that I only reveals a negligible amount of information about $K = G(I, A)$. In our model this means that it is computationally difficult to obtain a significant amount of information about K given I. In [LT03,VTD+03] the model is restricted to information theoretical security. The point of this paper is to show that by allowing computational security we obtain simple solutions..

See [LT03,VTD+03] for a deep discussion on contracting and revealing functions.

6.2 Physical Random Functions

Gassend et al. [GC+02a] introduced the concept of physical random functions. They defined a Physical Random Function (PUF)[10] to be a function that maps challenges to responses, that is embodied by a physical device, and that verifies the following properties:

1. Easy to evaluate: The physical device is easily capable of evaluating the function in a short amount of time.
2. Hard to characterize: From a polynomial number of plausible physical measurements (in particular, determination of chosen challenge-response pairs), an attacker who no longer has the device, and who can only use a polynomial amount of resources (time, matter, etc...) can only extract a negligible amount of information about the response to a randomly chosen challenge.

In this definition, the terms short and polynomial are relative to the size of the device, which is the security parameter. In particular, short means linear or low degree polynomial. The term plausible is relative to the current state of the art in measurement techniques and is likely to change as improved methods are devised.

In [R01] PUFs were referred to as Physical One Way Functions (POWFs), and realized using 3-dimensional micro-structures and coherent radiation. As explained in [GC+02a,GLC+] this terminology is confusing because PUFs do not match the standard meaning of one way functions [MvOV96]. A PUF is a one-way function in the sense that it is hard to reconstruct the physical system from challenge-response pairs. However, unlike a one-way function, a PUF does not require going from the response to the challenge to be hard. For a PUF,

[10] PUF actually stands for Physical Unclonable Function. It has the advantage of being easier to pronounce, and it avoids confusion with Pseudo-Random Functions.

all that matters is that going from a challenge to a response without using the device is hard.

In Figure 5 an optical PUF [R01] is embedded in a smartcard. The PUF plays the role of the physical system P in the profile matching model and is used for authentication and identification. During a secure bootstrapping phase in which Bob is in physical contact with the smartcard, Bob receives challenge response pairs. In Figure 5, C is such a challenge and B is its corresponding response. Here, C represents a laser beam characterized by its angle and frequency.

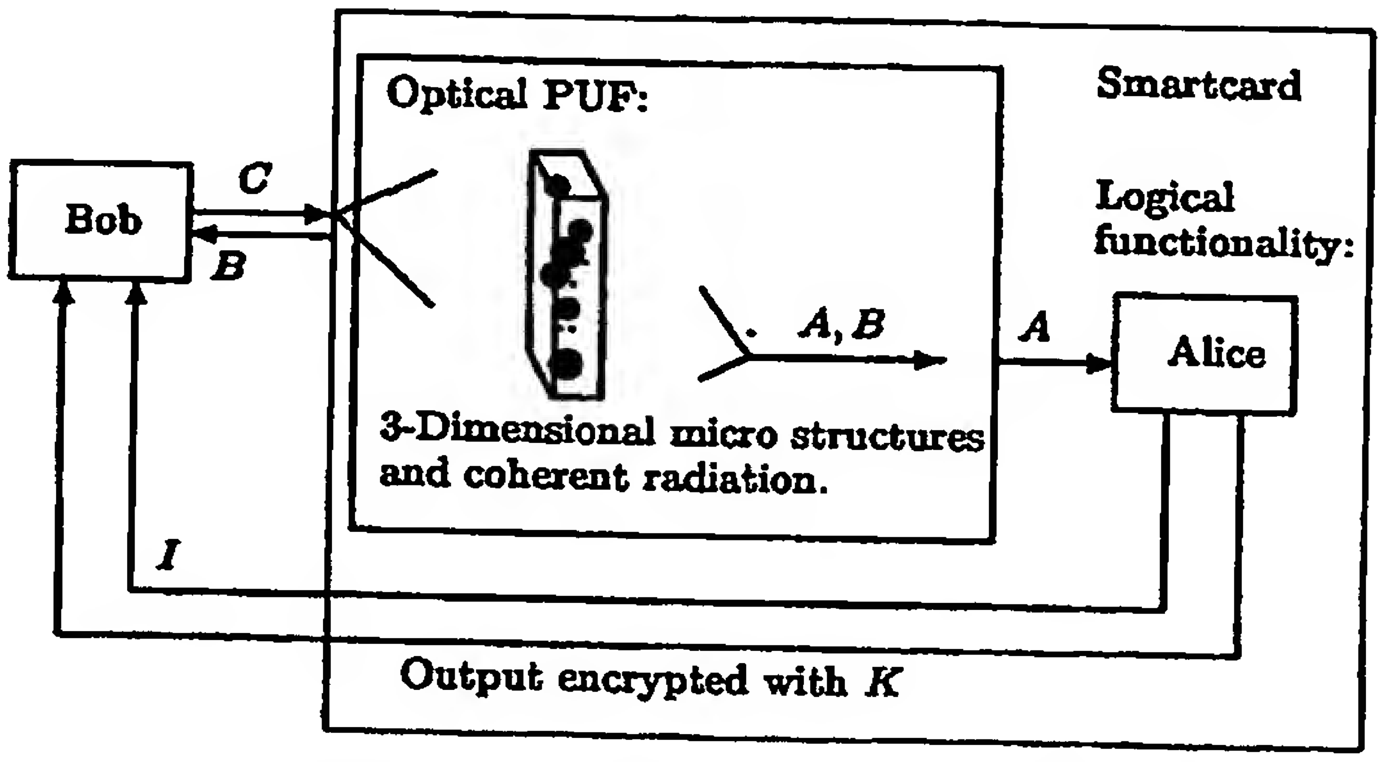

Fig. 5. Model for an optical PUF within a smartcard.

Some time after the bootstrapping phase one of Bob's ATM machines may want to securely communicate with the smartcard. Bob's ATM machine gives the challenge C to the smartcard and the physical system computes a corresponding response A. Due to environmental and measurement noise A and B may be different. The PUF represented by the physical system is not a function in the mathematical sense, it is a statistical process. Furthermore, an adversary may try to build a software model of the physical system and may try to extract useful information from other smartcards with similar PUFs. In conclusion, to create a secure function we need profile matching. For example, the chip within the smartcard (represented by Alice) generates an arbitrary key K and creates a code word W based on K and A. The chip computes the corresponding message I which is transmitted to Bob, who can recover K if the responses A and B are close enough to one another. An adversary who uses a software model and other smartcards to create a simulated response E cannot obtain any information about K because A and E will be far enough apart.

The advantage of using a PUF within a smartcard is that it is not clonable. After losing the smartcard you can still use it securely as soon as you find it again. Bob only identifies the PUF linked to the smartcard. To identify the owner of the smartcard we may want to merge the smartcard with biometrics.

After processing the incoming beam, we may extend the optical PUF's functionality to represent the outcoming pattern as a binary vector and to perform a post-hash. In Section 5.1 we discussed a profile matching protocol for such a physical system. The parameter p models the worst allowable environment and parameter q models the best attacking model (using multiple other smartcards and software modeling). Notice that protection against the best attacking model guarantees enough inter-PUF variations; PUFs are uniquely identified. If we are not allowed to perform a post-hash, then we need the solution described in Section 5.2.

In [GC+02a] silicon PUFs (SPUFs) are introduced. Individual integrated circuits (ICs) are identified based on a prior delay characterization of the IC. While IC's can be reliably mass-manufactured to have identical digital logic functionality, each IC is unique in its delay characteristics due to inherent variations in manufacturing across different dies, wafers, and processes. While digital logic functionality relies on timing constraints being met, different IC's with the exact same digital functionality will have unique behaviors when these constraints are not met, because their delay characteristics are different. In [GC+03] a key-card application is described and it is shown that there is enough inter-chip variation to reliably identify FPGA's. In [GLC+] the security is further analyzed. Software modeling based on machine learning algorithms leads to a factor higher bit error probability compared to the bit error probability due to environmental[11] and measurement noise. The solutions in Section 5.1 need in the order of $5 \cdot 10^5$ challenge response pairs. Since SPUFs need about $100ns$ to compute one challenge response pair, our solutions need about 1 second to generate a reliable and secure key.

Gassend et al. [GC+02b] defined a PUF to be Controlled (CPUF) if it can only be accessed via an algorithm that is physically linked to the PUF in an inseparable way (i.e., any attempt to circumvent the algorithm will lead to the destruction of the PUF). In particular this algorithm can restrict the challenges that are presented to the PUF and can limit the information about responses that is given to the outside world.

As explained in [GC+02b] control turns out to be the fundamental idea that allows PUFs to go beyond simple authenticated identification applications. CPUFs can be used to provide tamper-resistance, anonymous computation[12], and trusted third party computation[13] with applications in certified execution and software-licensing. Control enables these applications by trusting only a

[11] Temperature and voltage variations. There are no results on aging effects.

[12] Alice wants to run computations on Bob's computer, and wants to make sure that she is getting correct results. A certificate is returned with her results to show that they were correctly executed.

[13] Alice and Bob agree on a program to run. They run it on a chip that they both trust (identified by a PUF). They each provide the program with confidential information knowing that it won't leak the information.

single-chip processor that contains a PUF. In other words, the PUF contains the key material on which the security of these applications is be based[14].

CPUFs use pre-hashes, this avoids successful attacks by using other CPUFs to get information about the response. CPUFs use post-hashes, this makes model building impossible. Therefore, there is no threat of a malicious Eve. We only need to consider honest Eve's which are other CPUFs who should not be wrongly identified. This is the profile matching model in which Eve has no additional computational resources ($M = 0$). This means we can use any of our protocols.

7 Concluding remarks

We discussed, analyzed reliability and security of, and gave solutions for profile matching in the presence of an adversary. If the adversary has a finite amount of computing power then we may use one-way functions. The most practical application to date is physical random functions. Together with the presented techniques they can be made reliable and secure by using a practical solution based on one-way functions.

References

[AC93] R. Ahlswede and I. Csiszár. Common randomness in information theory and cryptography – Part I: secret sharing. *IEEE Trans. on Information Theory*, 39(4), p. 1121-1132, 1993.

[C97] C. Cachin. *Entropy measures and unconditional security in cryptology.* Ph. D. Thesis, ETH Zürich, Hartung-Gorre Verlag, Konstanz, 1997.

[CH77] A.B. Carleial and M.E. Hellman. A note on Wyner's wiretap channel. *IEEE Trans. on Information Theory*, p. 387–390, 1977.

[CW01] B. Chen and G.W. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 47(4), p. 1423-1443, 2001.

[CT91] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley, 1991.

[CK78] I. Csiszár and J. Körner. Broadcast channels with confidential messages. *IEEE Trans. on Information Theory*, 24(3), p. 339-348, 1978.

[vD97] M. van Dijk. *Secret key sharing and secret key generation.* Ph. D. Thesis, Technische Universiteit Eindhoven, 1997.

[EHM+00] C. Ellison, C. Hall, R. Milbert, and B. Schneier. Protecting secret keys with personal entropy. *Future Generation Computer Systems*, 16(4), p. 411-318, 2000.

[FK89] D.C. Feldmeier and P.R. Karn. UNIX password security - ten years later. In *Advances in Cryptology - Crypto'89*, LNCS 435, p. 44-63, 1989.

[GC+02a] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, 2002.

[14] We do not want to base the security on the difficulty of reading out digital keys stored in registers. Differential power analysis ...

[GC^{+}02b] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled Physical Random Functions. *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC'02)*, 2002.

[GC^{+}03] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Delay-based circuit authentication and applications. *Proceedings of the 2003 ACM Symposium on Applied Computing (SAC'03)*, 294-301, 2003.

[GLC^{+}] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas. Identification and authentication of integrated circuits. accepted for publication in Concurrency and Computation: Practice and Experience.

[GS99] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon codes and algebraic-geometry codes. *IEEE Trans. on Information Theory*, 45(6), p. 1757-1767, 1999.

[JS02] A. Juels and M. Sudan. A fuzzy vault scheme. *Proceedings of the 2002 IEEE International Symposium on Information Theory*, p. 408, 2002.

[JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. *6th ACM Conference on Computer and Communication Security*, p. 28-36, 1999.

[KJV^{+}03] R. Koetter, M. Jun, A. Vardy, and A. Ahmed. Efficient interpolation and factorization in algebraic soft-decision decoding of Reed-Solomon codes. *Proceedings of the 2003 IEEE International Symposium on Information Theory*, p. 365, 2003.

[KV00] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *Proceedings of the 2000 IEEE International Symposium on Information Theory*, p. 61, 2000. *38th Annual Allerton Conference on Communication, Control and Computing*, 2000. Full manuscript: citeseer.nj.nec.com/koetter00algebraic.html.

[KM77] J. Körner and K. Marton. General broadcast channels with degraded message sets. *IEEE Trans. on Information Theory*, 23(1), p. 60-64, 1977.

[LT03] J.-P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. *4th International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003.

[LvTvD03] S. Liu, H.C.A. van Tilborg, and M. van Dijk. *A practical protocol for advantage distillation and information reconciliation. Designs, Codes and Cryptography*, 30, p. 39-62, 2003.

[MS77] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[M83] J.L. Massey. A simplified treatment of Wyner's wire-tap channel. *Proc. 21st Annual Allerton Conf. on Communication, Control and Computing*, p. 268-276, 1983.

[M91] U.M. Maurer. Perfect cryptographic security from partially independent variables. *Proc. 23st Annual ACM Symposium on Theory of Computing*, p. 561-571, 1991.

[M93] U.M. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. on Information Theory*, 39, p. 733-742, 1993.

[MvOV96] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[MT79] R. Morris and K. Thompson. Password security: a case history. *Communications of the ACM*, 22, p. 594-597, 1979

[PV03] F. Parvaresh and A. Vardy. Multiplicity assignments for algebraic soft-decoding of Reed-Solomon codes. *Proceedings of the 2003 IEEE International Symposium on Information Theory*, p. 205, 2003.

[P80] P. Piret. Wire-tapping of a binary symmetric channel. *Philips J. Res. 35*, p. 251-258, 1980.

[R01] P.S. Ravikanth. *Physical one-way functions*. Ph. D. Thesis, Massachusetts Institute of Technology, 2001.

[VTD+03] E. Verbitskiy, P. Tuyls, D. Denteneer, and J.P. Linnartz. Reliable biometric authentication with privacy protection. *The IEEE Benelux Symp. on Inf. Theory*, Veldhoven, The Netherlands, 2003.

[W96] S.G. Wilson. *Digital Modulation and Coding*. Prentice Hall, Inc., 1996.

[W99] S. Wolf. *Information-theoretically and computationally secure key agreement in cryptography*. ETH dissertation No. 13138, ETH Zürich, 1999.

[W75] A.D. Wyner. The wire-tap channel. *Bell System Tech. J.*, 54, 1355-1387, 1975.

A Summary

1. Suppose that Alice receives $(X_1, \ldots, X_n)$ and Bob receives $(Y_1, \ldots, Y_n)$. Let $h(.)$ be a function which is easy to evaluate but for which its inverse is hard to compute. Then Alice may transmit $(h(X_1), \ldots, h(X_n))$ to Bob to create an erasures channel. That is, Bob computes $h(Y_i)$ and if $h(Y_i) \neq h(X_i)$ then Bob knows that $Y_i \neq X_i$ and he assigns an erasure to Y_i. If $h(Y_i) = h(X_i)$ then Bob knows that $Y_i = X_i$. The erasure information can be used by Bob in higher layer error correction schemes to reconstruct the complete vector $(X_1, \ldots, X_n)$.
2. As in 1, but now Bob computes a set U_i (possibly based on Y_i) and he computes $h(Z)$ for each $Z \in U_i$. Bob compares the evaluations $h(Z)$ with $h(X_i)$. If $X_i \in U_i$ then Bob will find a Z such that $h(Z) = h(X_i)$ (hence, $Z = X_i$) and Bob assigns Z to Y_i. If $X_i \notin U_i$ then Bob will not find a Z such that $h(Z) = h(X_i)$ and he assigns an erasure to Y_i.
3. In 1 or 2, Alice may encode $(h(X_1), \ldots, h(X_n))$ into a code word

   $$(h(X_1), \ldots, h(X_n), P_1, \ldots, P_{d-1})$$

   of some error correcting code, and Alice may transmit $(P_1, \ldots, P_{d-1})$ instead of $(h(X_1), \ldots, h(X_n))$. Upon receiving $(P_1, \ldots, P_{d-1})$, Bob computes $(h(Y_1), \ldots, h(Y_n), P_1, \ldots, P_{d-1})$ and he uses a decoding algorithm to reconstruct $(h(X_1), \ldots, h(X_n), P_1, \ldots, P_{d-1})$. Now Bob proceeds as described in 1 or 2.
4. In 1, 2, or 3, Bob may transmit to Alice the set S of indices i for which $h(Y_i) = h(X_i)$. Then Alice and Bob both know S and they can use $(X_i)_{i \in S} = (Y_i)_{i \in S}$ to extract or generate for example a shared secret key.
5. In 4 applied to 1 or 2, Alice and Bob may receive unordered sets $\{X_1, \ldots, X_n\}$ and $\{Y_1, \ldots, Y_n\}$ resp. instead of vectors $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_n)$ resp. Then, Alice first orders her set into a vector $(X_1, \ldots, X_n)$ and she proceeds as in 1 to transmit $((h(X_1), \ldots, h(X_n))$. Then Bob computes as in 2 the set S of indices i for which he found a Z with $h(Z) = h(X_i)$ (as in 2 he assigns such a Z to Y_i).

6. In 4 or 5, Bob may also transmit $K + \mathtt{Hash}(Y_i : i \in S)$. Alice receives $K + \mathtt{Hash}(Y_i : i \in S)$ and subtracts $\mathtt{Hash}(X_i : i \in S)$ leading to K. This procedure leads to a shared secret key K between Alice and Bob.
7. In 4 or 5, Alice may transmit $K + \mathtt{Hash}(X_i : i \in S)$ after having received S from Bob. Then, Bob receives $K + \mathtt{Hash}(X_i : i \in S)$ and he subtracts $\mathtt{Hash}(Y_i : i \in S)$ leading to K. This procedure leads to a shared secret key K between Alice and Bob.
8. Suppose that Alice receives $(X_1, \ldots, X_n)$ and Bob receives $(Y_1, \ldots, Y_n)$. To generate a secret key Alice encodes $(K_1, \ldots, K_k, X_1, \ldots, X_n)$ into a code word $(K_1, \ldots, K_k, X_1, \ldots, X_n, P_1, \ldots, P_{d-1})$ of some error correcting code, and Alice transmits $(P_1, \ldots, P_{d-1})$ to Bob. Bob constructs

   $$(?, \ldots, ?, Y_1, \ldots, Y_n, P_1, \ldots, P_{d-1})$$

   (where ? denotes an erasure) and uses a decoding algorithm to retrieve $(K_1, \ldots, K_k, X_1, \ldots, X_n, P_1, \ldots, P_{d-1})$. Then Alice and Bob both know

   $$(K_1, \ldots, K_k, X_1, \ldots, X_n)$$

   which they use to extract or distill a shared secret key (this is called privacy amplification).
9. In 8, we may use a RS code. For example, if the X_i and Y_i are bits we may group them into m-bit symbols representing elements in $GF(2^m)$ or we may regard each bit as a symbol 0 or 1 in $GF(2^m)$, and we use a RS code over $GF(2^m)$.
10. In 8, privacy amplification can be accomplished by left-multiplication of a random $(k+n) \times s$ matrix to distill s bits. Or Alice and Bob can use a hash function. Or, if RS codes are used, Alice and Bob distill $(K_1, \ldots, K_k)$ as the secret key.
11. The higher layer coding scheme which makes use of the erasure channel as described in 1, 2, 3, 4, 5, 6, or 7 can be the one as described in 8, 9, or 10.
12. The previous ideas (1 to 11) can be applied in biometrics, see Section 6.1, or can be applied to make PUFs reliable and secure, see Section 6.2.
13. In 12 for controlled PUFs we may adapt the GetResponse and GetSecret primitive to match our ideas. For example, both primitives need to implement decoding algorithms (in for example 8, 9, and 10) or extra output (set S in 4, 5, 6, and 7).
14. In 12 for PUFs used for identification or authentication, the PUF will identify or authenticate itself by proving (in a secure way) that it knows the shared secret key which was generated by our ideas.